

ATARI COMPUTER ENTHUSIASTS

3662 Vine Maple Dr. Eugene OR 97405

SEPTEMBER, 1983

Mike Dunn & Jim Bumpas, Editors

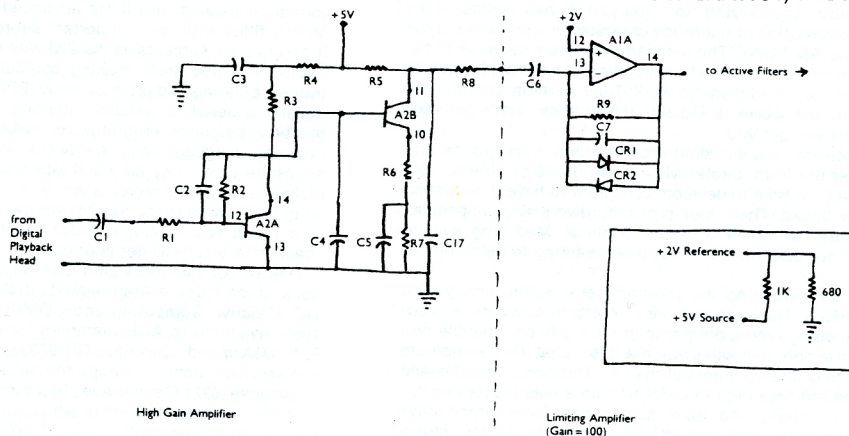
Antic
The Atari Magazine

Figure 1

ATARI 410 PROGRAM RECORDER
DIGITAL PLAYBACK CIRCUIT
SCHEMATIC DIAGRAM

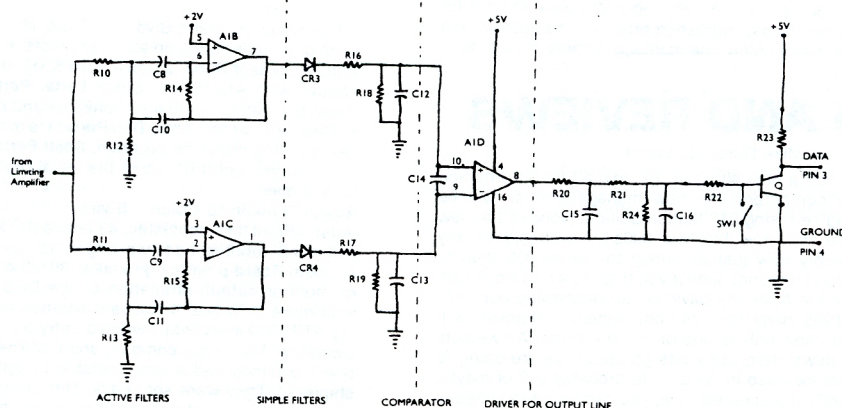
TAPE TOPICS

SCHEMATIC by Carl Evans



April 1983

qq



See CLOAD ERRORS by Steven Matern, pg15

POLICY STATEMENT

ATARI COMPUTER ENTHUSIASTS
EUGENE, OREGON
Kirt E. Stockwell
President

As we start another year of meetings and activities, I want to take this opportunity to restate some of the club's basic policies.

The primary purpose of the ACE is to promote intelligent use of the personal computer in the home and educational environment. We believe the computer can be used to augment traditional educational programs, as well as eventually becoming a major educational tool.

We also believe computer technology has reached the stage at which normal, everyday american families can genuinely benefit by the use of the personal computer. The speed and storage limitations previously relegating personal computers to the realm of 'expensive toy' have largely been overcome.

Without software, the computer is useless. There must be incentive for the creation, testing and sale of software. It takes lots of time and energy to develop programs to be sold. Publishers need a market for a product large enough to make a profit from the sale of software. Finally, local dealers need to be able to sell enough of each piece of software to justify tying up cash to maintain an item in inventory.

You have read in these pages we (spelled I) believe the software publishing companies are mistaken in their pricing structures. I have talked with several publishing firms and have heard some pretty lame excuses for exhorbitant pricing structures. For the time being we will have to live with the fact of high prices. It may help to realize the prices (theoretically) can drop as volume rises.

This brings us to the topic: PIRACY.

Piracy is a crime punishable by imprisonment or fine. The ATARI COMPUTER ENTHUSIASTS of EUGENE condemns piracy and, as always, will cooperate as fully as possible with authorities to eliminate pirates.

Many of you know Atari has retained a well known and effective private investigator specializing in piracy. He has successfully assisted in the prosecution of hundreds of pirates. He is known to me and others only as 'Mad Dog'. The nickname attaches because of his effectiveness and vindictiveness when it comes to pirates. His results reflect the enjoyment he finds in his work. Take warning: Do not consider piracy, it is not worth it. Do not think "they won't get me," because Mad Dog will get you.

Pirates include individuals, retail organizations, user groups and others. No matter the form, pirates violate the rights of others. Consider: You work a long time to develop a program to have it published and make a few bucks. Then your profit, hardwork and respect all disappears because of some pirate, a criminal. Mad Dog will be organizing a 24 hour BBS for those out there wanting to help combat piracy.

Aside from theft violating the programmer's rights, piracy also causes higher prices. The large number of computer owners is growing. This will encourage more programmers to make commercial programs because the potential sales volume is so large. Pirates operate to reduce the size of the commercial market. This reduces sales and profits. So prices are kept high in order to gain a reasonable profit.

Some pirates attempt to gain acceptance from prospective customers by implying they are affiliated with respected clubs, groups, or persons. Some even impersonate respected persons. This causes problems for the individual, group, or club. It also causes problems for the general public because they don't know who to trust.

Many local ACE members are developing software intended for publication. Several have already published programs. As authors and potential authors, we understand the damage pirates cause to a creator.

NEWS AND REVIEWS

by Mike Dunn, Co-Editor

Hope you have all had a nice summer. By now all the information about the new Atari computers and new management should be known to you all. Imagine losing 312 Million dollars. None of the new ones are out yet, as far as I know, but Atari 800s are available for very low prices; it is probably now just as cheap to get an 800 than to upgrade a 400. Although I have not seen it yet, there is a new book called "Free Software for the Atari" by David Heller describing our club, and we have been getting many inquiries from readers — thanks! I will review it when I get a copy, unless one of you do it first. As we continue to improve our newsletter, our costs go up, so we are going to have to have a modest increase in dues — to probably \$12 or maybe \$15, starting next month. If you renew now, you can save the increase — we are still one of the least expensive Atari Clubs.

We have received many nice books the last month. For general use, for those just familiar to BASIC as well as fairly advanced users, is a very nice book from Reston Publishing Co. (11480 Sunset Hills Rd., Reston, VA 22090). It is a new book on Atari Graphics called **Designs from Your Mind with Atari Graphics** by Tom Rowley (\$12.95). Written to help you become a computer artist, this book includes not only the usual material about all the Atari Graphic modes, character sets, PM Graphics, etc., but also much about composition of your art, animation, one and two point perspective, custom display lists, and other advanced topics. It even includes screen design worksheets for each graphic mode. A very nice book to have.

Reston has also come out with the first book in their new Software Integration Series, **The DIF File** by Donald Bell (\$15.95). This series is devoted to books explaining how to integrate various different softwares. The DIF file is the format now used by VisiCalc and many other programs. Developed by Bob Frankston of Software Arts, it is a software format allowing various types of software to exchange information. An interesting sidelight is that when VisiCalc first came out, it was available for only the Apple, and many bought the Apple to get the program. The original Apple version did not have the DIF format; and when the Atari version was released it was one of the first to use this format. The Apple version has been upgraded several times since; but unfortunately, the Atari version is the same. This book explains how to make and use DIF files, how to integrate them between various pieces of software, and how to write your own software to use the DIF files generated by VisiCalc. If you need to know this, you will learn all you need in this well written and clearly explained book.

Another new Reston book is **ComputerTown** (\$12.95) by Liza Loop et al. It is a "How to" manual on how to start a non-profit computer town for your community as started in Menlo Park, California, to bring "Computer literacy to YOUR community". Besides the expected chapters on why and how to organize, it has chapters on fund-raising, writing proposals and grants, plans on building learning centers and easy to build furniture. It even includes a section on how to keep your equipment running.

ELCOMP (53 Redrock Lane, Pomona, CA 91766), the German based publisher, has just released a translation of **Hackerbook for your Atari Computer** by H.C. Wagner. This book has an incredible amount of information packed into it for advanced programmers and "hackers". Mostly filled with "very important subroutines in machine language", it covers such subjects as the CIO with machine language subroutines and how to use them; making bootable programs and bootable cartridges; building and programming an EPROM burner; changing devices; making a serial or parallel interface using the joystick ports with machine language programs to make them work; and much other machine language and hardware information just not available anywhere else. They also sell kits and circuit boards if you want to make the hardware projects easier. I'm not sure of the cost, but most all of ELCOMP's products are very low priced, so I expect this book is also much too inexpensive for all that it contains! If you are a "hacker", be sure to get your copy!

One of our members (Jim Carr) in a nearby town has produced a book which indexes Atari related articles from ACE, A.N.A.L.O.G., ANTIC, Creative Computing and COMPUTE! from April 1981 to March 1983. Available to ACE members for only \$6 from Valley Soft, 2660 S.W. DeArmond, Corvallis, OR 97333, this is a wonderful time saver and very well done — I hope the author keeps it up.

Synapse (5221 Central Ave., Richmond, CA 94804) has just announced a new integrated series of application software which are inter-compatible. This series will include **SynText**, **SynFile +**, **Syn-Trend**, **SynCalc**, **Syn-Stock**, **Syn-Comm**, **Syn-Stat**, **Syn-Tax** (not a pun!) and **Syn-Mail**. These will all use the Axlon 128K board, both 80 column boards, and double-density drives! I can hardly wait to review these. Look at Lotus 1-2-3!!

LJK (7852 Big Bend Blvd., St. Louis, MO 63119) has just announced large price reductions on all its products. For example, their excellent word-processor ROM is down from \$200 to \$100, making it very competitive with AtariWriter. Since **Letter Perfect** comes with a printer driver generator, it actually is cheaper and more powerful now. I call it a "Best Buy" at this time. **DataPerfect** is also \$99 now, and they have a new spelling checking program, **Spell Perfect** for \$79.95 which I have not seen yet, but if it's up to the usual high standards of LJK, should be a winner.

Austin-Franklin 80 Column Board

(Austin-Franklin Associates, 43 Grove St., Ayer, Ma 01432, \$279)

Several months ago, we began to see ads for the new Austin-Franklin Board promising to allow 80*25 character Screen, RGB color or monitor output, emulation of the DEC VT100, and compatibility with most software. Because it seemed to be an ideal companion to my ATR 8000 and CP/M, I wanted to try one. After a very long wait, ACE Computer finally got some in, and I borrowed one. Unfortunately, the one I obtained had a very unstable image, as did others in the first shipment. They were sent back, and the replacements had the same problem. In the instructions it says you should not adjust anything, especially a certain thing. Since this seemed the most likely one to

have been out of adjustment, we moved it a very tiny bit and everything became perfect. Now I could finally begin testing the unit. The picture on my Taxan amber monitor is very nice — the RGB output now requires a special adaptor cable costing \$69.95. The unit works by plugging it into your last slot on the 800, so you need a 32K, 48K or 64K board to free up the slot. To use the 80 Columns, you plug a ROM into the right slot; this emulates an ADM-3 and does not take up memory. The DEC V100 software is not yet available (\$39.95). There is no requirement for software calls, as in the BIT-3. When we first received the unit, it did not work with the ATR-8000, but after a few calls, the company quickly designed a very nice ROM for the right slot, which works very well for the ATR-8000 and CP/M. If you have the CP/M, this is a very worth-while investment.

Many commercial programs will not work with this board in 80 columns. The board uses addresses in \$D500 and \$D600 pages, and also will not work with any programs using illegal entry points. None of the present word-processors, data-bases, VisiCalc, etc. will work. Most languages such as BASIC, MicroSoft BASIC, FORTH, and most assemblers, especially the ones from OSS work fine. Any software designed for BIT-3 will not work. Synapse's new integrated software and data-base series will work, as well as a new version of Letter Perfect, but I don't know when it will be available. I will keep you posted on this fine product as more software becomes available. Right now, I recommend it only for the ATR-8000 owners with CP/M (very highly!!) and to avoid programmers, or people who hook into mainframes a lot and use the Atari as a Terminal — the MicroBits Modem and Terminal program works very well with it.

ATR 8000 News

Speaking of the ATR 8000, the other exciting news is the release of MYDOS, an Atari DOS now available allowing you to use double-sided drives with an upgrade ROM (not quite available), 80 track disk drives, etc. The RS-232 handler is just about ready to ship also. SWP's terminal cartridge costs \$29.95. Owners contact the company or locally L.J. Knoll at ACE Computers (343-5191) for details. The Co-Power-88, turning the Atari into a IBM-PC type computer is also working well, but only with "generic" MS-DOS software — not specific IBM-PC hardware dependent software. This seems to be the case with all the IMP-PC clones.

NEW FROM ACE

There are new several new items available from ACE. George Suet-sugu, 45-602 Apuapu St., Kaneohe, HI 96744, who bound all of the issues of ACE from the beginning to June, 1982 for \$12, has now made available **Best of ACE Book Volume II**, July, 1982 to June, 1983 for the same price, and he includes a nice little surprise for all orders. For overseas members, please include extra postage for about 1 pound airmail.

The second disk of 1983, **Best of ACE disk vol 6** is now available from the ACE Exchange Library 374 Blackfoot, Eugene, OR 97404. Run by Aaron and Ron Ness, we are now able to supply high quality tapes made by the new interface/Stereo tape deck as reviewed by Larry Gold. Disk or tape is \$15, and covers from June to Sept of 1983, including the latest **TinyText2** and the **PM Graphics Maker** in this issue. Incidentally, when requesting the ACE Exchange library list from the Ness', please enclose 25 cents (50 cents overseas) in stamps — not an envelope as before!

The latest **Educational Disk #4** from our E.R.A.C.E. group is also now available for \$10 — it is the best yet; please read the article by Robert Browning in this issue.

ERACE

ACE Educational disk No.4

ERACE's educational disk #4 is now ready. We are very happy with the quality of the programs on this disk, and are sure you will find them entertaining as well as educational.

The nine programs cover a wide range of subjects:

MATH WARS by Andrew Acks, is a very interesting approach to math drill. (See the July/August newsletter for listing and review);

ODDEVEN, PLURALS, NOUNS, and VERBS, by Wayne Real, are four odd man out games on numbers and parts of grammar. (Again see the last issue for more information);

QUIZPRO, and COPYQUIZ, by Roy Dugan, are two programs written to be used on a system with a cassette player and 16K. These programs were written so a teacher or parent with only a cassette player and 16K could write and give true-false or multiple choice tests. Instructions for using this program will be on the disk. This program is very special because it can be used on smaller systems.

From the Spokane, Washington users group we received the following programs for the new disk:

HANGQUOT is a variation of hangman in which you guess not only words but whole quotations;

GEOQUIZ is a guessing game for the computer. You think of a country and the computer tries to guess the right one. To narrow down its options it asks you questions which really make you think about the country;

MOOMIND is a variation of Master Mind. You try to guess a hidden number, but here the computer will give you clues using bulls and cows. It's a little different and a good logic game.

Well, now that disk #4 is done, we can start on #5. But we are out of programs. If you have a program to share, whether it's a product of your own sweat and time, or a copy of a good public domain program, send it in. We all will like to see it.

We also have 3 disks from Softswap which we are making available to interested persons. Write for more information on the educational programs on these disks.

We want to thank ALL of those who have made ERACE disk #4 possible. Thank you and on to #5.

LOOKING AT LOGO

by Ruth Ellsworth

Our family had the opportunity to preview the new ATARI LOGO recently. We do not presently have either the documentation or cartridge, so the technical information and programs to compare PILOT and LOGO will have to wait until next month. We can, however, give our first impressions after spending a couple of weeks experimenting with the language.

The documentation package which will be included in the LOGO package is excellent. It includes a booklet teaching LOGO programming through the use of turtle graphics, a reference booklet, and a summary pamphlet of commands. The booklet teaching LOGO programming is extremely well written and extensively illustrated. Due to my busy summer schedule, my ten and twelve year olds tackled the booklet on their own and did not require any help at all from me to complete it well before I had the time to begin. In our opinion, even those unfamiliar with computers or computer languages will find the documentation easy to use and understand.

ATARI LOGO turtle graphics are superb, and the most exciting thing to come to our home in a long time. The children liked the LOGO turtle better than the PILOT turtle. One of our sons has not been very interested in programming, but even he was enchanted by it. I think there are several reasons for this: The LOGO turtles are cute little fellows one can see. There are four of them which can move at different speeds and have collisions if desired (anyone whose children suffer from Star Raider's wrist or Galaxian's glare can understand the significance of this discovery!). And the increments in LOGO allow the children to use larger numbers to move the turtle. This last item may not seem like much, but the children were delighted to make the turtle go 100 steps and still be able to see it on the screen.

LOGO is to PASCAL and LISP as PILOT is to BASIC. There are no line numbers in LOGO, and modules, called procedures, are defined and called within a program. It has the ability to redefine functions as in FORTH or PASCAL. For someone who is not trained in programming or mathematics, to me it had a comfortable almost verbal feel once I began to work with it.

Now comes the \$64 question (or should I say \$99 question). What should I buy, LOGO or PILOT? Next month I should have a LOGO package and will do a detailed comparison. For now, the answer to this question in simplistic terms depends upon what one needs. If one plans or needs to prepare standard curriculum type materials, or anything requiring much matching in a program, ATARI PILOT is the language to buy. If, on the other hand, one needs to write programs dealing with symbol manipulation, or complex data structures, LOGO is the language of choice because those structures do not have to be encoded in mathematical terms or strings. This List capability which makes LOGO less mathematical, also seems to make it harder to learn than PILOT, although LOGO and PILOT graphics are equally easy. Thus, if one is pressed for time, PILOT has an initial advantage in my opinion.

For me, the answer is I want to have both LOGO and PILOT. ATARI has done an excellent job in preparing and packaging both languages, and if one can afford them, they are well worth the money.

CHARTING A COURSE IN PILOT

by Ruth Ellsworth

In our home we like to think writing a program is a little like writing a book. Every chapter covers a set of ideas or events, and has a beginning and an end. Programs are made of "subroutines", called modules in PILOT. Like the chapters of a book, each contains an idea (routine or procedure) the program uses, and must have a beginning and an end.

A fun way to learn about and practice programming with modules is through writing adventure type programs. I am not going to include a complete adventure program with this article (they can be as endless as the programmer chooses). Instead, I am going to use adventure type modules like the ones my children write to demonstrate how to use the U:, J:, M:, TY:, NY:, EY:, EN:, and JM: commands. I hope this will encourage creative learning on the part of other beginners as they have for our family.

Each module in a program has an * followed by the name of the module. The name line is NOT numbered. After the name, the body of the module which follows is numbered by line, and concluded with an E: It is important to put the E: at the end of each module because the computer will run through the modules which follow, or if it is not told to jump elsewhere. The E: also makes programs neater and easier to read.

Most of the PILOT programs appearing in the ACE Newsletter do not start with a module. Before the program begins it is necessary to set any counters or constants (things which will not change), and call the graphics mode in a graphics program so there will be no stray cursor on the screen. The title may be included either in a module, or as part of the beginning of the program before any modules are called.

In the program which follows one will note the modules do not follow in the order in which they will be used. For example, *RANDOM is listed before *TITLE, AND *HELP; although *TITLE, AND *HELP are used by the program first. In order for PILOT to run as fast as possible, those modules which will be used most often should be placed near the beginning of the listing. Every time a module is called, PILOT starts from the top and looks through the listing until it finds the module called.

Both the U: and J: commands are followed by the *name of a module. The U: command stands for Use Module, and the J: commands stands for Jump (to Module). The important difference between them is that U: will return the program to the module from which it came. For example, see the *ROOM1 module which U:*DOORS before it proceeds to ask the adventurer his choice. A U: can only repeat eight times before it returns an error message.

The J: does not return the program to the module which called it. We like to think of the J: as resetting the internal counter each time it is called so a module can be reused indefinitely. At first my children wondered why they could not just use the J: command. Can you figure out why the U: command is more useful in some cases? (Hint: look at the use of the *DOORS module. What happens if J:*DOORS is used?)

The M: command checks to match an answer (input) from the user to an answer (input) wanted in the program. Numbers, words (strings), and even bytes in the computer's memory can be checked using the M: command. (If you have the ACE Christmas PILOT disk, the FLOWCOL program illustrates the use of the M: command to match bytes.) If more than one item is to be matched they must be separated by commas or a |.

There are several useful commands which go with the M: command. The TY: command stands for Type if Yes there was a match, and the TN: command stands for Type if No match. Similarly EY: stands for End if Yes there was a match, and the EN: command stands for End if No match.

The command we use most often with the M: command is the JM: command which stands for Jump on Match (to a named module). Each JM: module listed must be paired to an item in the M: statement. The JM: command matches the items in the M: in the order given. The first item is matched by the first module name listed in the JM: statement, and so forth.

Writing programs in modules is good programming practice. Like plotting a course in flying an airplane, each program should be planned ahead. Good pilots take the time to figure out what their program will do and how they wish to travel through it. Not only does modular programming save time debugging, but it saves much typing if modules are saved individually. This way they can be used as is or modified for new programs.

TINY TEXT 2.0

One of the nicest things about Tiny Text is it can easily be modified to suit one's needs. I have added a few more features I feel Tiny Text needs. First of all, get out your copy of Tiny Text 1.1 and type in the lines as listed in Listing 1. The new features added to Tiny Text are:

CTRL-D can be imbedded in the text in exactly the same manner as CTRL-M is. CTRL-D is used to set the line spacing — useful for single spacing quotes in a document otherwise double spaced. The number after CTRL-D is the spacing. If you want single spacing, type [CTRL-D][1][!]. If you want double spacing [CTRL-D][2][!], and so on.

While editing, the trigger button on the joystick may also be used to get you to the top of the text. Simply push the trigger once and you're at the bottom; again and you're at the top.

When you are faced with the SET FORMAT CONTROLS table, two more parameters await your adjustment. The second new one, LINE INTERVAL, sets the line spacing as explained above. The first, PAGE NUMBERING, gives the option of page numbering. A 0 here means no, a 1 means yes. If you put a 1 there, you will be asked for a page heading. If you simply type the return key, only "Page x" will be right justified on the top of each new page. If you enter a heading (e.g. "Rat Problem in Alberta"), it will be printed just before a slash and the page number (e.g. "Rat Problem in Alberta/Page 4"). You can also opt for using single sheets. If you do, Tiny Text will sound a tone and wait for you to insert the sheet and push a key before it starts printing again.

At any time during displaying or printing you may freeze the computer by pushing any key. The border will change color to let you know that you have stopped the presses. To resume, simply push any key.

In order to speed up printing, screen output is disabled by POKEing 559 with 0. This speeds up things considerably, many sources say up to a 30% speed increase can be realized. This could be used to speed up any programs when the user doesn't need to see the screen. Screen output is turned back on by POKEing 559 with its original value (usually 34).

If you have a disk drive, enter in Listing 2. This adds a DOS section to Tiny Text. I recommend you enter it in as a separate program, and then merge it with Tiny Text for reasons I'll get to a little further down.

When you first enter the TINYDOS portion of Tiny Text by pushing START from within the program, you will be given an initial screen of files on the disk. An inverse video letter before the file indicates it is locked; normal video indicates an unlocked file. If not all the files fit on one screen, pushing the space bar will get you the next batch. OPTION will exit you back to Tiny Text. SELECT, followed by a letter, will lock or unlock the file, depending on its state at the time. START followed by a letter will delete the chosen file. You will need to push the Y key to confirm each deletion, however. Pushing START and SELECT simultaneously followed by any key will get you into the rename portion, which requires you to enter the following string: "D:the old filename and extension, the new filename and extension".

You may notice the DOS routine seems to take its time. The reason is a tradeoff had to be made between speed and memory. Since we want as much memory for text as possible, I chose to sacrifice speed.

Listing 3 contains changes which make Listing 2 into a run alone program which you could have automatically run (by using an AUTORUN.SYS file) when you boot up with a disk. The only functional change made is you can push OPTION, followed by a key, to RUN any BASIC SAVED program.

While Tiny Text is starting to get fairly untiny, I feel the power it has gained is very much worth the slight loss of memory.

— Dale Lutz
Holden, Alberta, Canada



More PPTOI programs for
your enjoyment

410 *ROOM4
420 T:¶
430 T:THIS IS THE TREASURE ROOM. IN THE CHEST YOU WILL FIND GOLD AND A KEY TO THE WAY OUT.
440 T:CONGRATULATIONS! COME AGAIN.
450 E:
460 *DOORS [THESE ARE ALL CONTROL CHARACTERS

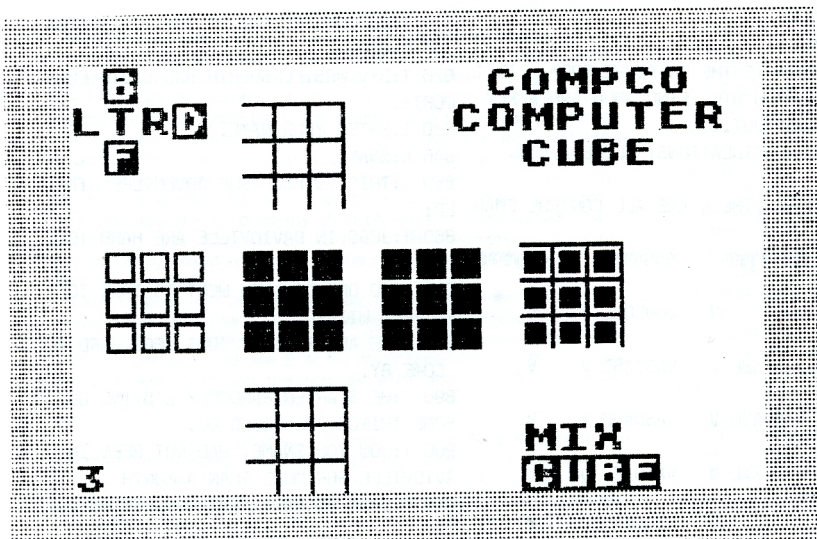
470 T:	MMMMMMMMMM	MMMMMMMMMM	MMMMMMMMMM
MM			
480 T:	V	V	VQWEQWE V
V			
490 T:	VQWEQWE V	VASDASD V	V
V			
500 T:	VASDASD V	VASDASD V	V
V			
510 T:	VZXCZXC V	VASDASD V	V
V			
520 T:	V	TV	VASDASDTV
TV			
530 T:	VQWEQWE V	VASDASD V	V
V			
540 T:	VASDASD V	VASDASD V	V
V			
550 T:	VZXCZXC V	VASDASD V	V
V			
560 T:	V	V	VASDASD V
V			
570 T:	V	V	VASDASD V
V			
580 T:	V	V	VZXCZXC V
V			
590 T:	NNNNNNNNNNNN	NNNNNNNNNNNN	NNNNNNNNNNNN
NNN			
600 E:			
610 *FRIEND			
620 T:¶			
, 630 T:			
640 T:			
650 T:			
660 T:			
670 T:			
680 T:			
690 T:			
700 T:			
RO 710 T:			
720 T:	HERE IS A FRIEND TO HELP YOU.		
E	WE WILL LET YOU CHOOSE ONE OF TWO ROOMS.		
.	ONE OF THE ROOMS HAS THE TREASURE.		
730 T:	WHICH ROOM WILL YOU CHOOSE?		
, R	R 2		
740 A:			
750 M:	1,2		
760 JM:	*ROOM1,*ROOM4		
770 E:			
N 780 *HELP			
790 GR:	QUIT		
800 C:	@B755=2		

```

810 T:WELCOME TO ONE OF THE..CATACOMBS
  OF DAVID
820 T:BY..MRS.ELLSWORTH AND DAVID ELLS
  WORTH.
830 T:ENTER YOUR NAME.
840 A:$NAME
850 T:THIS IS HOW YOUR ADVENTURE START
  ED:
860 T:JOBS IN DAVIDVILLE ARE HARD TO C
  OME BY.
870 T:SO ONE DAY YOU WENT TO TALK TO T
  HE TOWN WIZARD.
880 T:HE AGREED THAT JOBS WERE HARD TO
  COME BY.
890 T:HE STARTED RANDOMLY LISTING OUT
  SOME THINGS YOU COULD DO.
900 T:NOW YOU,$NAME, HAD NOT BEEN IN D
  AVIDVILLE FOR MORE THAN A MONTH
910 T:WHEN HE SAID "THE DUNGEON OF DAV
  ID"
920 T:YOU FOUND YOURSELF IN A ROOM YOU
  HAD NEVER SEEN BEFORE.
930 T:WATCH OUR FOR MONSTERS WHILE YOU
  HUNT FOR GOLD AND THE WAY OUT.
940 PA:100
950 J:*RANDOM
960 E:
970 *TITLE
980 R:TITLE PAGE FOR PROGRAM
990 C:@B1373=16
1000 C:@B1374=2
1010 WRITE:S,¶
1020 U:*TITLE1
1030 E:
1040 *TITLE1
1050 POS:2,5
1060 C:@B755=2
1070 WRITE:S,DAVID'S DUNGEON
1080 R:WAIT UNTIL START IS HIT
1090 PA:60
1100 J(@B53279<>6):*TITLE2
1110 J(@B53279=6):*HELP
1120 E:
1130 *TITLE2
1140 POS:2,5
1150 C:@B755=4
1160 WRITE:S,DAVID'S DUNGEON
1170 R:WAIT UNTIL START IS HIT
1180 PA:60
1190 J(@B53279<>6):*TITLE1
1200 J(@B53279=6):*HELP
1210 E:

```


CUBES by Sydney Brown



Solve the cube puzzle

```

1 REM *****
2 REM **      ACE NEWSLETTER      **
3 REM **      3662 VINE MAPLE DR   **
4 REM **      EUGENE, OR 97405    **
5 REM **      $10 YEAR             **
6 REM *****
7 REM **      CUBE                 **
8 REM **      by                   **
9 REM **      Sydney Brown Sept 1983 **
   *****
10 ? "I":DIM P(55):OPEN #1,4,0,"K:"
100 POKE 106,PEEK(106)-2
101 GRAPHICS 18:POKE 710,140:POKE 708,
14:POKE 709,222:POKE 711,58:A=PEEK(106
)*256:FOR B=0 TO 511
102 IF B>455 AND B<472 THEN READ D:POK
E A+B,D:NEXT B
105 POKE A+B,PEEK(57344+B):NEXT B:POKE
756,PEEK(106)
120 MP=PEEK(88)+256*PEEK(89)
200 POSITION 12,0: ? #6;"compco":POSITI
ON 11,1: ? #6;"computer":POSITION 13,2
: ? #6;"cube":POSITION 1,0: ? #6;"B":PO
SITION 0,1: ? #
201 POSITION 13,10: ? #6;"MIX":POSITIO
N 13,11: ? #6;"CUBE":POSITION 1,2: ? #6;
"f":GOSUB 900:J=3:POSITION 0,11: ? #6;"
3 "
202 IF PEEK(53279)=5 THEN J=J+1:IF J>2
0 THEN J=3
203 POSITION 0,11: ? #6;J;" ":IF PEEK(
53279)=6 THEN 300
205 FOR W=1 TO 49:NEXT W:GOTO 202
210 POSITION 13,10: ? #6;"FACE":POSITIO
N 13,11: ? #6;"DIR.";
```

```

212 POKE 764,255
215 IF PEEK(53279)=6 THEN 201
216 IF PEEK(764)=255 THEN 215
220 GET #1,Z:Z=Z-64:IF Z=2 OR Z=4 OR Z
=6 OR Z=12 OR Z=18 OR Z=20 THEN 225
224 GOTO 215
225 POSITION 13,10
226 IF Z=12 THEN ? #6;"left":Z=1:GOTO
235
227 IF Z=2 THEN ? #6;"BACK":Z=2:GOTO 2
35
228 IF Z=20 THEN ? #6;"TOP ":Z=3:GOTO
235
229 IF Z=6 THEN ? #6;"FRNT":Z=4:GOTO 2
35
230 IF Z=18 THEN ? #6;"right":Z=5:GOTO
235
231 IF Z=4 THEN ? #6;"down":Z=6:GOTO 2
35
235 IF PEEK(53279)=6 THEN 201
236 IF PEEK(764)=255 THEN 235
237 GET #1,V:V=V-64:IF V=1 OR V=3 THEN
240
239 GOTO 235
240 POSITION 13,11:IF V=1 THEN ? #6;"a
c<=":Z=Z+6
241 IF V=3 THEN ? #6;"c1>=";
245 ON Z GOSUB 310,320,330,340,350,360
,410,420,430,440,450,460
299 GOTO 210
300 FOR W=1 TO J:R=INT(RND(0)*12)+1:ON
R GOSUB 310,320,330,340,350,360,410,4
20,430,440,450,460
301 NEXT W:GOTO 210
310 I=PEEK(P(1)):POKE P(1),PEEK(P(7)):
POKE P(7),PEEK(P(9)):POKE P(9),PEEK(P(
3)):POKE P(3),I
311 I=PEEK(P(2)):POKE P(2),PEEK(P(4)):
POKE P(4),PEEK(P(8)):POKE P(8),PEEK(P(
6)):POKE P(6),I
```

```

312 I=PEEK(P(10)):POKE P(10),PEEK(P(54
)):POKE P(54),PEEK(P(28)):POKE P(28),P
EEK(P(19)):POKE P(19),I
313 I=PEEK(P(13)):POKE P(13),PEEK(P(51
)):POKE P(51),PEEK(P(31)):POKE P(31),P
EEK(P(22)):POKE P(22),I
314 I=PEEK(P(16)):POKE P(16),PEEK(P(48
)):POKE P(48),PEEK(P(34)):POKE P(34),P
EEK(P(25)):POKE P(25),I
319 RETURN
320 I=PEEK(P(10)):POKE P(10),PEEK(P(16
)):POKE P(16),PEEK(P(18)):POKE P(18),P
EEK(P(12)):POKE P(12),I
321 I=PEEK(P(11)):POKE P(11),PEEK(P(13
)):POKE P(13),PEEK(P(17)):POKE P(17),P
EEK(P(15)):POKE P(15),I
322 I=PEEK(P(1)):POKE P(1),PEEK(P(19))
:POKE P(19),PEEK(P(37)):POKE P(37),PEE
K(P(46)):POKE P(46),I
323 I=PEEK(P(2)):POKE P(2),PEEK(P(20))
:POKE P(20),PEEK(P(38)):POKE P(38),PEE
K(P(47)):POKE P(47),I
324 I=PEEK(P(3)):POKE P(3),PEEK(P(21))
:POKE P(21),PEEK(P(39)):POKE P(39),PEE
K(P(48)):POKE P(48),I
329 RETURN
330 I=PEEK(P(20)):POKE P(20),PEEK(P(22
)):POKE P(22),PEEK(P(26)):POKE P(26),P
EEK(P(24)):POKE P(24),I
331 I=PEEK(P(19)):POKE P(19),PEEK(P(25
)):POKE P(25),PEEK(P(27)):POKE P(27),P
EEK(P(21)):POKE P(21),I
332 I=PEEK(P(16)):POKE P(16),PEEK(P(9)
):POKE P(9),PEEK(P(30)):POKE P(30),PEE
K(P(37)):POKE P(37),I
333 I=PEEK(P(17)):POKE P(17),PEEK(P(6)
):POKE P(6),PEEK(P(29)):POKE P(29),PEE
K(P(40)):POKE P(40),I
334 I=PEEK(P(18)):POKE P(18),PEEK(P(3)
):POKE P(3),PEEK(P(28)):POKE P(28),PEE
K(P(43)):POKE P(43),I
339 RETURN
340 I=PEEK(P(29)):POKE P(29),PEEK(P(31
)):POKE P(31),PEEK(P(35)):POKE P(35),P
EEK(P(33)):POKE P(33),I
341 I=PEEK(P(28)):POKE P(28),PEEK(P(34
)):POKE P(34),PEEK(P(36)):POKE P(36),P
EEK(P(30)):POKE P(30),I
342 I=PEEK(P(54)):POKE P(54),PEEK(P(45
)):POKE P(45),PEEK(P(27)):POKE P(27),P
EEK(P(9)):POKE P(9),I
343 I=PEEK(P(53)):POKE P(53),PEEK(P(44
)):POKE P(44),PEEK(P(26)):POKE P(26),P
EEK(P(8)):POKE P(8),I
344 I=PEEK(P(52)):POKE P(52),PEEK(P(43
)):POKE P(43),PEEK(P(25)):POKE P(25),P
EEK(P(7)):POKE P(7),I
349 RETURN
```



```

350 I=PEEK(P(38)):POKE P(38),PEEK(P(40))
:POKE P(40),PEEK(P(44)):POKE P(44),P
EEK(P(42)):POKE P(42),I
351 I=PEEK(P(37)):POKE P(37),PEEK(P(43))
:POKE P(43),PEEK(P(45)):POKE P(45),P
EEK(P(39)):POKE P(39),I
352 I=PEEK(P(18)):POKE P(18),PEEK(P(27))
:POKE P(27),PEEK(P(36)):POKE P(36),P
EEK(P(46)):POKE P(46),I
353 I=PEEK(P(15)):POKE P(15),PEEK(P(24))
:POKE P(24),PEEK(P(33)):POKE P(33),P
EEK(P(49)):POKE P(49),I
354 I=PEEK(P(12)):POKE P(12),PEEK(P(21))
:POKE P(21),PEEK(P(30)):POKE P(30),P
EEK(P(52)):POKE P(52),I
359 RETURN
360 I=PEEK(P(47)):POKE P(47),PEEK(P(49))
:POKE P(49),PEEK(P(53)):POKE P(53),P
EEK(P(51)):POKE P(51),I
361 I=PEEK(P(46)):POKE P(46),PEEK(P(52))
:POKE P(52),PEEK(P(54)):POKE P(54),P
EEK(P(48)):POKE P(48),I
362 I=PEEK(P(12)):POKE P(12),PEEK(P(45))
:POKE P(45),PEEK(P(34)):POKE P(34),P
EEK(P(1)):POKE P(1),I
363 I=PEEK(P(11)):POKE P(11),PEEK(P(42))
:POKE P(42),PEEK(P(35)):POKE P(35),P
EEK(P(4)):POKE P(4),I
364 I=PEEK(P(10)):POKE P(10),PEEK(P(39))
:POKE P(39),PEEK(P(36)):POKE P(36),P
EEK(P(7)):POKE P(7),I
369 RETURN
410 I=PEEK(P(1)):POKE P(1),PEEK(P(3)):
POKE P(3),PEEK(P(9)):POKE P(9),PEEK(P(
7)):POKE P(7),I
411 I=PEEK(P(2)):POKE P(2),PEEK(P(6)):
POKE P(6),PEEK(P(8)):POKE P(8),PEEK(P(
4)):POKE P(4),I
412 I=PEEK(P(10)):POKE P(10),PEEK(P(19))
:POKE P(19),PEEK(P(28)):POKE P(28),P
EEK(P(54)):POKE P(54),I
413 I=PEEK(P(13)):POKE P(13),PEEK(P(22))
:POKE P(22),PEEK(P(31)):POKE P(31),P
EEK(P(51)):POKE P(51),I
414 I=PEEK(P(16)):POKE P(16),PEEK(P(25))
:POKE P(25),PEEK(P(34)):POKE P(34),P
EEK(P(48)):POKE P(48),I
419 RETURN
420 I=PEEK(P(10)):POKE P(10),PEEK(P(12))
:POKE P(12),PEEK(P(18)):POKE P(18),P
EEK(P(16)):POKE P(16),I
421 I=PEEK(P(11)):POKE P(11),PEEK(P(15))
:POKE P(15),PEEK(P(17)):POKE P(17),P
EEK(P(13)):POKE P(13),I
422 I=PEEK(P(1)):POKE P(1),PEEK(P(46)):
:POKE P(46),PEEK(P(37)):POKE P(37),PEE
K(P(19)):POKE P(19),I
423 I=PEEK(P(2)):POKE P(2),PEEK(P(47)):
:POKE P(47),PEEK(P(38)):POKE P(38),PEE
K(P(20)):POKE P(20),I

```

```

424 I=PEEK(P(3)):POKE P(3),PEEK(P(48))
:POKE P(48),PEEK(P(39)):POKE P(39),PEE
K(P(21)):POKE P(21),I
429 RETURN
430 I=PEEK(P(20)):POKE P(20),PEEK(P(24))
:POKE P(24),PEEK(P(26)):POKE P(26),P
EEK(P(22)):POKE P(22),I
431 I=PEEK(P(19)):POKE P(19),PEEK(P(21))
:POKE P(21),PEEK(P(27)):POKE P(27),P
EEK(P(25)):POKE P(25),I
432 I=PEEK(P(16)):POKE P(16),PEEK(P(37))
:POKE P(37),PEEK(P(30)):POKE P(30),P
EEK(P(9)):POKE P(9),I
433 I=PEEK(P(17)):POKE P(17),PEEK(P(40))
:POKE P(40),PEEK(P(29)):POKE P(29),P
EEK(P(6)):POKE P(6),I
434 I=PEEK(P(18)):POKE P(18),PEEK(P(43))
:POKE P(43),PEEK(P(28)):POKE P(28),P
EEK(P(3)):POKE P(3),I
439 RETURN
440 I=PEEK(P(29)):POKE P(29),PEEK(P(33))
:POKE P(33),PEEK(P(35)):POKE P(35),P
EEK(P(31)):POKE P(31),I
441 I=PEEK(P(28)):POKE P(28),PEEK(P(30))
:POKE P(30),PEEK(P(36)):POKE P(36),P
EEK(P(34)):POKE P(34),I
442 I=PEEK(P(54)):POKE P(54),PEEK(P(9))
:POKE P(9),PEEK(P(27)):POKE P(27),PEE
K(P(45)):POKE P(45),I
443 I=PEEK(P(53)):POKE P(53),PEEK(P(8))
:POKE P(8),PEEK(P(26)):POKE P(26),PEE
K(P(44)):POKE P(44),I
444 I=PEEK(P(52)):POKE P(52),PEEK(P(7))
:POKE P(7),PEEK(P(25)):POKE P(25),PEE
K(P(43)):POKE P(43),I
449 RETURN
450 I=PEEK(P(38)):POKE P(38),PEEK(P(42))
:POKE P(42),PEEK(P(44)):POKE P(44),P
EEK(P(40)):POKE P(40),I
451 I=PEEK(P(37)):POKE P(37),PEEK(P(39))
:POKE P(39),PEEK(P(45)):POKE P(45),P
EEK(P(43)):POKE P(43),I
452 I=PEEK(P(18)):POKE P(18),PEEK(P(46))
:POKE P(46),PEEK(P(36)):POKE P(36),P
EEK(P(27)):POKE P(27),I
453 I=PEEK(P(15)):POKE P(15),PEEK(P(49))
:POKE P(49),PEEK(P(33)):POKE P(33),P
EEK(P(24)):POKE P(24),I
454 I=PEEK(P(12)):POKE P(12),PEEK(P(52))
:POKE P(52),PEEK(P(30)):POKE P(30),P
EEK(P(21)):POKE P(21),I
459 RETURN
460 I=PEEK(P(47)):POKE P(47),PEEK(P(51))
:POKE P(51),PEEK(P(53)):POKE P(53),P
EEK(P(49)):POKE P(49),I
461 I=PEEK(P(46)):POKE P(46),PEEK(P(48))
:POKE P(48),PEEK(P(54)):POKE P(54),P
EEK(P(52)):POKE P(52),I
462 I=PEEK(P(12)):POKE P(12),PEEK(P(1))
:POKE P(1),PEEK(P(34)):POKE P(34),PEE

```

```

K(P(45)):POKE P(45),I
463 I=PEEK(P(11)):POKE P(11),PEEK(P(4))
:POKE P(4),PEEK(P(35)):POKE P(35),PEE
K(P(42)):POKE P(42),I
464 I=PEEK(P(10)):POKE P(10),PEEK(P(7))
:POKE P(7),PEEK(P(36)):POKE P(36),PEE
K(P(39)):POKE P(39),I
469 RETURN
900 ZZ=0:A=1:B=5:C=122:GOSUB 910:A=5:B
=1:C=217:GOSUB 910:A=5:B=5:C=89:GOSUB
910
902 A=5:B=9:C=249:GOSUB 910:A=9:B=5:C=
121:GOSUB 910:A=13:B=5:C=218:GOSUB 910
909 RETURN
910 COLOR C:FOR Y=B TO B+2:FOR X=A TO
A+2:PLOT X,Y:ZZ=ZZ+1:P(ZZ)=MP+(20*Y+X)
:NEXT X:NEXT Y:RETURN
1000 DATA 0,254,254,254,254,254,254,25
4,0,254,130,130,130,130,130,254

```

UTILITIES

by Shane Rolin

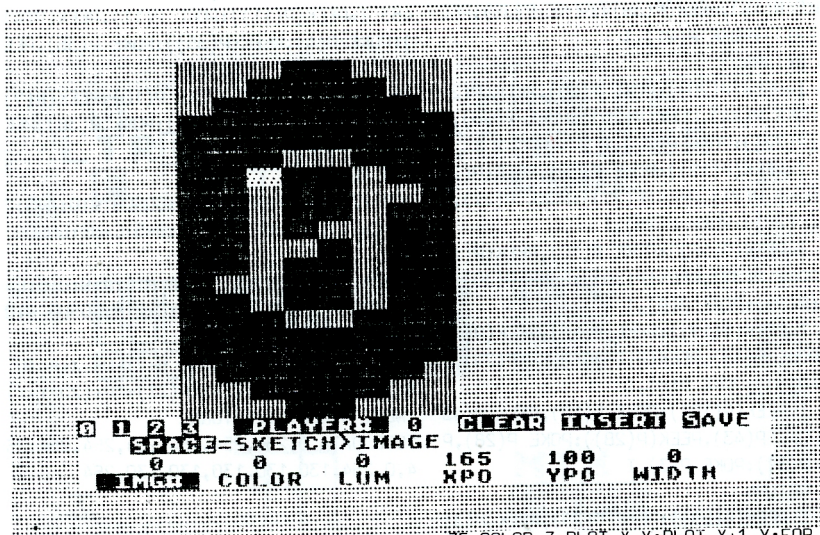
Useful Programs for MicroPainter Files

```

1 REM *****
2 REM ** ACE NEWSLETTER **
3 REM ** 3662 VINE MAPLE DR **
4 REM ** EUGENE, OR 97405 **
5 REM ** $12 YEAR **
6 REM *****
100 REM *-_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*
105 REM
110 REM MICROPAINTER PICTURE LOADER
115 REM (C)1983 C-SOFTWARE
120 REM 48k W/DISK req.
125 REM
127 REM BY SHANE ROLIN of the
130 REM -PITTSBURGH_ATARI COMPUTER-
132 REM -----ENTHUSIASTS-----
134 REM
135 REM AND A.C.E. EUGENE 9/83
140 REM *-_*-*_*-*_*-*_*-*_*-*_*-*_*-*_*
200 DIM MACHINE$(31),FILENAME1$(20),FI
LENAME2$(20),DIRECT$(1000):FILENAME1$=
"D:"
210 TRAP 470
220 GOSUB 370
230 OPEN #2,4,0,"K"
240 A=PEEK(708):B=PEEK(709):C=PEEK(710)
:D=PEEK(712):MODE=7:SCRNUM=7680:E=8:X
Y=13
250 GOTO 470 GOTO page 12
260 GOSUB 360
270 FILENAME1$(3,19)=FILENAME2$

```


Stan Ockers



*Make and save your own
PM Graphics*

```

1 REM *****
2 REM ** ACE NEWSLETTER **
3 REM ** 3662 VINE MAPLE DR **
4 REM ** EUGENE, OR 97405 **
5 REM ** $12 YEAR **
6 REM *****
7 REM ** PM HELPER 2 **
8 REM ** by **
9 REM ** Stan Ockers Sept 1983 **
10 DIM S$(6),IM(19),COLR(3),LUM(3),XPO
    (3),YPO(3),WID(3),IMG(3),F$(15)
12 GOSUB 10000:GRAPHICS 3
20 XPO(0)=PEEK(XPO0):XPO(1)=PEEK(XPO1)
:XPO(2)=PEEK(XPO2):XPO(3)=PEEK(XPO3):Y
PO(0)=PEEK(YPO0):YPO(1)=PEEK(YPO1)
21 YPO(2)=PEEK(YPO2):YPO(3)=PEEK(YPO3)
22 IMG(0)=PEEK(IMG0)/20:IMG(1)=PEEK(IMG
G1)/20:IMG(2)=PEEK(IMG2)/20:IMG(3)=PEE
K(IMG3)/20
23 FOR J=0 TO 3:C9=PEEK(1694+J):COLR(J
)=INT(C9/16):LUM(J)=C9-COLR(J):NEXT J
24 FOR J=0 TO 3:WID(J)=PEEK(1698+J):NE
XT J
25 POKE 764,255:X=10:Y=6
26 REM * Labels in lines 26-27 in inve
rse as appropriate *
27 PLR=0:POKE 712,6:POKE 752,1:POKE 6
56,0:POKE 657,0:?" 0 1 2 3 PLAYER#
";PLR;" CLEAR INSERT SAVE";
28 POKE 656,1:POKE 657,3:?"SPACE=SKET
CH>IMAGE";:SEL=1:GOSUB 300:GOSUB 350:P
OKE 53277,3:POKE 559,62:GOSUB 360
29 REM ** CURSOR **
30 LOCATE X,Y,Z
35 COLOR 3:PLOT X,Y:PLOT X+1,Y:FOR J=1
TO 10:NEXT J:COLOR Z:PLOT X,Y:PLOT X+
1,Y
39 REM ** CHANGE IF TRIG PRESSED **
40 IF STRIG(0)=0 AND Z=1 THEN COLOR 2:
PLOT X,Y:PLOT X+1,Y:GOSUB 2000:GOSUB 1
300
45 IF STRIG(0)=0 AND Z=2 THEN COLOR 1:
PLOT X,Y:PLOT X+1,Y:GOSUB 2200:GOSUB 1
300
49 REM ** MOVE ACCORDING TO JOYSTICK **
50 S=STICK(0)
60 IF S=14 AND Y>0 THEN Y=Y-1
70 IF S=13 AND Y<19 THEN Y=Y+1
80 IF S=7 AND X<20 THEN X=X+2
90 IF S=11 AND X>6 THEN X=X-2
94 REM ** CHECK FOR KEY PRESSED **
95 KEY=PEEK(764)
100 IF KEY=255 THEN 30:REM * NO KEY, L
OOP BACK *
102 IF KEY=62 THEN GOSUB 400:GOTO 25:R
EM * 'S' for save *
110 IF KEY=7 AND SEL<6 THEN SEL=SEL+1:
GOSUB 300:REM * Next selection right *
112 IF KEY=6 AND SEL>1 THEN SEL=SEL-1:
GOSUB 300:REM * Next selection left *
114 IF KEY=50 THEN PLR=0:GOSUB 350:RE
M * Player 0 *
115 IF KEY=31 THEN PLR=1:GOSUB 350:RE
M * Player 1 *
116 IF KEY=30 THEN PLR=2:GOSUB 350:RE
M * Player 2 *
117 IF KEY=26 THEN PLR=3:GOSUB 350:RE
M * Player 3 *
118 POKE 656,0:POKE 657,19:?"PLYR
120 IF KEY<>14 THEN 130:REM * Not up a
rrow *
125 ON SEL GOSUB 210,220,230,240,250,2
60:GOSUB 350
130 IF KEY<>15 THEN 150:REM * Not down
arrow *
135 ON SEL GOSUB 270,275,280,285,290,2
95:GOSUB 350
150 IF KEY=118 THEN GOSUB 370:REM * CL
EAR *
160 IF KEY=119 THEN GOSUB 360:REM * PL
YR>SKETCH 'INSERT' *
198 IF KEY=33 THEN GOSUB 1400:REM * SK
ETCH>PLYR 'SPACE' *
200 POKE 764,255:GOTO 30
209 REM * Raise or Lower Values Subrou
tines *
210 IF IMG(PLYR)>10 THEN RETURN
212 IMG(PLYR)=IMG(PLYR)+1:GOSUB 1403:R
ETURN
220 IF COLR(PLYR)>14 THEN RETURN
222 COLR(PLYR)=COLR(PLYR)+1:GOSUB 480
224 RETURN
230 IF LUM(PLYR)>12 THEN RETURN
232 LUM(PLYR)=LUM(PLYR)+2:GOSUB 480
234 RETURN
240 IF XPO(PLYR)>254 THEN RETURN
242 XPO(PLYR)=XPO(PLYR)+1:POKE 1668+PL
YR,XPO(PLYR):RETURN
250 IF YPO(PLYR)<1 THEN RETURN
252 YPO(PLYR)=YPO(PLYR)-1:POKE 1684+PL
YR,YPO(PLYR):RETURN
260 IF WID(PLYR)=0 THEN WID(PLYR)=1:GO
TO 264
262 IF WID(PLYR)=1 THEN WID(PLYR)=3
264 POKE 53256+PLYR,WID(PLYR):POKE 169
8+PLYR,WID(PLYR)
266 RETURN
270 IF IMG(PLYR)<1 THEN RETURN
272 IMG(PLYR)=IMG(PLYR)-1:GOSUB 1403:R
ETURN
275 IF COLR(PLYR)<1 THEN RETURN
277 COLR(PLYR)=COLR(PLYR)-1:GOSUB 480
279 RETURN
280 IF LUM(PLYR)<2 THEN RETURN
282 LUM(PLYR)=LUM(PLYR)-2:GOSUB 480
284 RETURN
285 IF XPO(PLYR)<1 THEN RETURN
286 XPO(PLYR)=XPO(PLYR)-1:POKE 1668+PL
YR,XPO(PLYR):RETURN
290 IF YPO(PLYR)>254 THEN RETURN
292 YPO(PLYR)=YPO(PLYR)+1:POKE 1684+PL
YR,YPO(PLYR):RETURN
295 IF WID(PLYR)=3 THEN WID(PLYR)=1:GO
TO 298
297 IF WID(PLYR)=1 THEN WID(PLYR)=0
298 GOTO 264
299 REM * PRINT VALUES LEGEND *
300 POKE 656,3:POKE 657,1:?" IMG# CC
LOR LUM XPO YPO WIDTH";
319 REM * All data in lines 321-326 ir
inverse chars *
320 RESTORE 320+SEL:READ S$:POKE 656,3
:POKE 657,1+6*(SEL-1):?" S$;

```



```

321 DATA IMG#
322 DATA COLOR
323 DATA LUM
324 DATA XPO
325 DATA YPO
326 DATA WIDTH
349 REM * PRINT VALUES LINE *
350 POKE 656,2:POKE 657,4: IMG(PLYR);
" ":POKE 657,10: COLR(PLYR);" ":POK
E 657,16: LUM(PLYR);" ";
352 POKE 657,21: XPO(PLYR);" ":POKE
657,27: YPO(PLYR);" ":POKE 657,34:
? WID(PLYR);
354 RETURN
359 REM * PLYR IMAGE > SKETCH AREA *
360 P=PMPG*256+256+IMG(PLYR)*20:FOR J=
0 TO 19:A=PEEK(P+J):IM(J)=A:POSITION 6
,J:W=256:FOR K=1 TO 8
362 W=W/2:IF A>W THEN A=A-W: #6:CHR$(
2):CHR$(2):GOTO 366
364 ? #6:CHR$(1):CHR$(1);
366 NEXT K:NEXT J:POKE 1676+PLYR,IMG(P
LYR)*20:RETURN
369 REM * CLEAR SKETCH AREA *
370 FOR I=0 TO 19:IM(I)=0:NEXT I:X=10:
Y=6
372 REM ** PRINT SKETCH AREA **
374 COLOR 1:FOR J=0 TO 19:PLOT 6,J:DRA
WTO 21,J:NEXT J
376 RETURN
399 REM * Save Routine *
400 A=USR(ADR(OUTVBI$)):POKE 656,1:POK
E 657,2: "FULL FILENAME FOR SAVE";:PO
KE 764,255:INPUT F$
410 GOSUB 500:GOSUB 600:LIST F$,10000,
10600:GOSUB 10320:GRAPHICS 3:RETURN
479 REM * Color Change *
480 C9=16*COLR(PLYR)+LUM(PLYR):POKE 70
4+PLYR,C9:POKE 1694+PLYR,C9:RETURN
499 REM * Re-enter lines to save image
*
500 GRAPHICS 0:POSITION 2,4:FOR J=0 TO
5:P=PMPG*256+256+20*J: 10420+10*J;"
DATA ":FOR K=0 TO 18
510 ? PEEK(P+K):",":NEXT K: PEEK(P+1
9):NEXT J: "CONT"
520 POSITION 2,0:POKE 842,13:STOP
530 POKE 842,12
540 GRAPHICS 0:POSITION 2,4:FOR J=6 TO
11:P=PMPG*256+256+20*J: 10420+10*J;"
DATA ":FOR K=0 TO 18
550 ? PEEK(P+K):",":NEXT K: PEEK(P+1
3):NEXT J: "CONT"
560 POSITION 2,0:POKE 842,13:STOP
570 POKE 842,12
580 RETURN

599 REM * Re-enter lines to save vario
us values *
600 GRAPHICS 0:POSITION 2,4: "10360 D
ATA 0,0,0,0,";:FOR J=1668 TO 1678: P
EEK(J):",":NEXT J: PEEK(1679)
610 ? "10365 DATA 0,0,0,0,";:FOR J=168
4 TO 1700: PEEK(J):",":NEXT J: PEEK
(1701): "CONT"
620 POSITION 2,0:POKE 842,13:STOP
630 POKE 842,12
640 RETURN
999 REM ** DELAY SUBR. **
1000 FOR I=1 TO 100:NEXT I:RETURN
1399 REM * Sketch>Image *
1400 PM=PMPG*256:FOR I=0 TO 19:POKE PM
+256+I+20*IMG(PLYR),IM(I):NEXT I
1402 REM * Make sure PM image updated
*
1403 POKE 1676+PLYR,240:FOR J=1 TO 20:
NEXT J:POKE 1676+PLYR,IMG(PLYR)*20
1405 RETURN
1999 REM ** ADD IN POWER OF 2 **
2000 IM(Y)=IM(Y)+INT(2^(8-(X-4)/2)+0.1
):RETURN
2199 REM ** SUB OUT POWER OF 2 **
2200 IM(Y)=IM(Y)-INT(2^(8-(X-4)/2)+0.1
):RETURN
9999 REM * PM Module *
10000 DIM MVPM$(48):RESTORE 10050:FOR
J=1 TO 48:READ A:MVPM$(J,J)=CHR$(A):N
EXT J
10010 A=USR(ADR(MVPM$))
10050 DATA 104,169,0,141,47,2,165,106,
133,204,56,233,8,133,206,133,106
10060 DATA 141,7,212,169,0,133,203,133
,205,168,162,8,177,203,145,205,169,0,1
45,203,136
10070 DATA 208,245,198,204,198,206,202
,208,238,96
10100 DIM VBI$(124):RESTORE 10150:FOR
J=1 TO 124:READ A:VBI$(J,J)=CHR$(A):N
EXT J
10150 DATA 216,162,0,173,156,6,24,105,
1,133,206,105,3,133,204,169,0,133,203,
189,128,6
10160 DATA 221,132,6,208,85,189,136,6,
221,140,6,208,8,189,144,6,221,148,6,24
0,59,189,140,6
10170 DATA 157,136,6,189,152,6,141,157
,6,188,144,6,169,0,145,203,200,206,157
,6,208,248
10180 DATA 189,152,6,141,157,6,160,0,1
89,140,6,133,205,189,148,6,133,203,177
,205,145,203,200
10190 DATA 206,157,6,208,246,189,148,6
,157,144,6,230,204,232,224,4,208,162,7
6,98,228
10200 DATA 189,132,6,157,128,6,157,0,2
08,24,144,234

10300 DIM INVBI$(10):RESTORE 10330:FOR
J=1 TO 10:READ A:INVBI$(J,J)=CHR$(A):
NEXT J
10310 DIM OUTVBI$(10):RESTORE 10340:FO
R J=1 TO 10:READ A:OUTVBI$(J,J)=CHR$(A
):NEXT J
10320 VBI=ADR(VBI$):HI=INT(VBI/256):LO
=VBI-256*HI:INVBI$(3,3)=CHR$(LO):INVBI
$(5,5)=CHR$(HI)
10330 DATA 104,160,0,162,0,169,7,76,92
,228
10340 DATA 104,160,98,162,228,169,7,76
,92,228
10342 GRAPHICS 0:POKE 752,1: "INITIAL
IZING..."
10350 RESTORE 10360:FOR J=1664 TO 1701
:READ A:POKE J,A:NEXT J
10360 DATA 0,0,0,0,165,175,185,195,0,2
0,40,60,0,20,40,60
10365 DATA 0,0,0,0,100,100,100,100,20,
20,20,20,152,0,0,0,0,0,0,0,0
10370 POKE 1692,PEEK(106):XPO0=1668:XP
01=1669:XPO2=1670:XPO3=1671
10380 YPO0=1684:YPO1=1685:YPO2=1686:YP
03=1687:IMG0=1676:IMG1=1677:IMG2=1678:
IMG3=1679
10386 FOR J=0 TO 3:POKE 704+J,PEEK(169
4+J):NEXT J
10388 FOR J=0 TO 3:POKE 53256+J,PEEK(1
698+J):NEXT J
10400 PMPG=PEEK(1692):RESTORE 10420:FO
R J=0 TO 239:READ A:POKE PMPG*256+256+
J,A:NEXT J
10410 POKE 623,1:A=USR(ADR(INVBI$)):RE
TURN
10420 DATA 231,195,129,0,0,24,36,38,36
,44,52,36,100,36,24,0,0,129,195,231
10430 DATA 231,195,129,0,0,24,8,8,8,8,
8,8,8,8,28,0,0,129,195,231
10440 DATA 231,195,129,0,0,60,36,4,4,1
2,24,48,32,36,60,0,0,129,195,231
10450 DATA 231,195,129,0,0,24,36,4,4,8
,4,4,4,36,24,0,0,129,195,231
10460 DATA 231,195,129,0,0,12,20,20,36
,36,62,4,4,4,4,0,0,129,195,231
10470 DATA 231,195,129,0,0,28,32,32,32
,24,4,4,4,36,60,0,0,129,195,231
10480 DATA 231,195,129,0,0,28,32,32,32
,56,36,36,36,36,60,0,0,129,195,231
10490 DATA 231,195,129,0,0,60,4,4,8,8,
16,16,16,16,16,0,0,129,195,231
10500 DATA 231,195,129,0,0,24,36,36,36
,60,36,36,36,36,24,0,0,129,195,231
10510 DATA 231,195,129,0,0,60,36,36,36
,60,4,4,4,36,60,0,0,129,195,231
10520 DATA 231,195,129,0,0,46,42,42,42
,42,42,42,42,42,46,0,0,129,195,231
10530 DATA 231,195,129,0,0,36,36,36,36
,36,36,36,36,36,0,0,129,195,231

```


PM POKE Source Listing by Stan Ockers

```

10 ; PM MOVEMENT BY POKES IN VBI
20 ;
0000 30      *= $0680
0680 40 OLDX  *= *+4 (1664)
0684 50 NEWX  *= *+4 (1668)
0688 60 OLDIMG *= *+4 (1672)
068C 70 NEWIMG *= *+4 (1676)
0690 80 OLDY  *= *+4 (1680)
0694 90 NEWY  *= *+4 (1684)
0698 0100 SIZE *= *+4 (1688)
069C 0110 PMPG *= *+1 (1692)
069D 0120 SIZECNT *= *+1 (1693)
0000 0130 HPOSPO = $0000 (53248)
00CB 0140 IND1L = $CB (203)
00CC 0150 IND1H = $CC (204)
00CD 0160 IND2L = $CD (205)
00CE 0170 IND2H = $CE (206)
E462 0180 XITVBI = $E462
0190 ;
069E 0200      *= $00
0000 08 0210 CLD ; DEC won't work right without this!
0001 A200 0220 LDH $00 ; Player index
0003 809C06 0230 LDA PMPG ; Set pointer..
0006 18 0240 CLC
0007 6901 0250 ADC $01
0009 85CE 0260 STA IND2H
000B 6903 0270 ADC $03 ; Other pointer..
000D 85CC 0280 STA IND1H ; to PLYR0 area
000F A900 0290 PLYR LDA $00
0011 85CB 0300 STA IND1L
0013 808006 0310 LDA OLDX,X ; Has X position..
0016 008406 0320 CMP NEWX,X ; changed?
0019 0055 0330 BNE UPDATX ; Yes
001B 808806 0340 LDA OLDIMG,X ; Has image..
001E 008C06 0350 CMP NEWIMG,X ; changed?
0021 0008 0360 BNE NUIMG ; Yes
0023 809006 0370 LDA OLDY,X ; Has Y position..
0026 009406 0380 CMP NEWY,X ; changed?
0029 F038 0390 BEQ NEXT ; No, skip
002B 808C06 0400 NUIMG LDA NEWIMG,X ; Update old..
002E 908806 0410 STA OLDIMG,X
0031 809806 0420 LDA SIZE,X ; Keep track of..
0034 809D06 0430 STA SIZECNT ; loops
0037 8C9006 0440 LDY OLDY,X ; Offset into plyr area
003A A900 0450 LDA $00
003C 91CB 0460 ERASE STA (IND1L),Y ; Erase old plyr
003E CB 0470 INY
003F CE9006 0480 DEC SIZECNT ; More?
0042 00F8 0490 BNE ERASE ; Yes
0044 809806 0500 LDA SIZE,X
0047 809D06 0510 STA SIZECNT ; Count loops again
004A A000 0520 LDY $00 ; No offset this time
004C 808C06 0530 LDA NEWIMG,X ; Point to..
004F 85CD 0540 STA IND2L ; new image
0051 809406 0550 LDA NEWY,X ; Fix vert pos..
0054 85CB 0560 STA IND1L ; in plyr area
0056 81CD 0570 FILL LDA (IND2L),Y ; Transfer image..
0058 91CB 0580 STA (IND1L),Y ; to plyr area
005A CB 0590 INY
005B CE9006 0600 DEC SIZECNT ; More?
005E 00F6 0610 BNE FILL ; Yes
0060 809406 0620 LDA NEWY,X ; Update Y pos.
0063 909006 0630 STA OLDY,X
0066 E6CC 0640 NEXT INC IND1H ; Next plyr area
0068 E8 0650 INX ; Next plyr
0069 E004 0660 CPX $04 ; Finished?
006B 00A2 0670 BNE PLYR ; No
006D 4C62E4 0680 JMP XITVBI ; Back to Atari VBI
0070 808406 0690 UPDATX LDA NEWX,X ; Update X pos..
0073 908006 0700 STA OLDX,X ; in shadow
0076 900000 0710 STA HPOSPO,X ; and register
0079 18 0720 CLC ; go to next plyr
007A 90EA 0730 BCC NEXT

```

Move Top Source Listing by Stan Ockers

```

10 ; ROUTINE TO MOVE 8 PAGES AND MAKE ROOM FOR PM GRAPHICS
20 ;
=022F 30 SOMCTL = $022F ;(559)
=0231 40 SOLSTH = $0231 ;(561)
=006A 50 RAMTOP = $6A ;(106)
=0407 60 PMBASE = $0407 ;(54279)
=00CB 70 FRPNTL = $CB ;(203)
=00CD 80 TOPNTL = $CD ;(205)
90 ;
0000 0100      *= $00
0000 68 0110 PLA ; Discard count
0001 A900 0120 LDA $00 ; Turn off DMA
0003 802F02 0130 STA SOMCTL ; for speed
0006 A56A 0140 LDA RAMTOP ; Highest page
0008 85CC 0150 STA FRPNTL+1 ; into pointer
000A 38 0160 SEC
000B E908 0170 SBC $08 ; 8 pages lower
000D 85CE 0180 STA TOPNTL+1 ; into other pointer
000F 856A 0190 STA RAMTOP ; Update RAMTOP
0011 800704 0200 STA PMBASE ; and PM base
0014 A900 0210 LDA #0 ; LD bytes of
0016 85CB 0220 STA FRPNTL ; pointers
0018 85CD 0230 STA TOPNTL
001A A8 0240 TAY ; Start with index 0
001B A208 0250 LOX $08 ; Eight pages
001D 81CB 0260 LOOP LDA (FRPNTL),Y ; Move bytes..
001F 91CD 0270 STA (TOPNTL),Y ; down
0021 A900 0280 LDA $00 ; Zero out top
0023 91CB 0290 STA (FRPNTL),Y ; eight pages
0025 88 0300 DEY ; Next index
0026 00F5 0310 BNE LOOP ; Not finished page
0028 C6CC 0320 DEC FRPNTL+1 ; Down a page
002A C6CE 0330 DEC TOPNTL+1
002C CA 0340 DEX ; More?
002D 00EE 0350 BNE LOOP ; Yes
002F 60 0360 RTS

```

MICROPAINTER PICTURE LOADER

by SHANE ROLIN of C-SOFTWARE (C)1983 (48k disk)

This program is a useful utility for people who want to view their Micropainter files. The second program is included for persons who need a fast routine to load in Micropainter files in their own programs. Here is a break-down of the structure of Micropainter Picture Loader.

LINES:

100-140 REM STATEMENTS
200-360 MAIN PROGRAM—TO LOAD INTO MEMORY AND
DISPLAY THE PICTURE
370-460 THE STRING TO LOAD IN THE PICTURE IS READ FROM
DATA LINES TO CHARACTERS IN A STRING
470-560 OPENING TITLE SCREEN WITH COMMANDS
570-680 CHECKS THE START, SELECT, AND OPTION KEYS AND
MOVES THE CURSOR
690-730 DETERMINES WHAT THE USER SELECTED
740-840 ASKS THE USER FOR AN INPUT FILENAME
850-940 THIS IS THE SUBROUTINE TO DISPLAY THE DIREC-
TORY OF DISK DRIVE 1
950-1050 ASKS THE USER WHAT FILE TO DELETE

Listing starts on page 7

TINYTEXT2 by Dale Lutz

*a Major Update of TinyText
originally by Stan Ockers and
modified by Jim Carr*

TinyText2 Change Listing 1 by Dale Lutz

```
12 DIM SP$(100):FOR I=1 TO 100:SP$(I,I)
  )=" ":NEXT I
115 MODE=1:LINT=1:PG=0
370 IF STRIG(0)=0 AND P=LEN(T$)-240 TH
  EN P=241:GOTO 300
375 IF STRIG(0)=0 THEN P=LEN(T$)-240:P
  OKE 207,0
505 POKE ST+24,65:POKE 559,34
560 POKE 764,255:POSITION 20,1:?"q":P
  OSITION 0,1:ON OP GOTO 2000,290,594,15
  00,594
590 REM
594 POSITION 0,1:?"SET FORMAT CONTROL
  S":POSITION 0,6:?"LINE LEFT IN- TAB
  PAGE FORM PAGE LINE"
595 ? "SIZE MARG DENT STOP SIZE FEED N
  UM. INT.":?" ":LL:?", ":LM:?", ":IND
  :?", ":TAB:?", ":PS:?", ":FF:?", ":
  600 ? PG:?", ":LINT:POSITION 0,8:INPU
  T LL,LM,IND,TAB,PS,FF,PG,LINT:P=240:PA
  GE=1
610 IF PG THEN ? :?"ENTER PAGE HEADIN
  G (/Page xx IS ADDED)":INPUT #2,I$:IF
  LEN(I$) THEN I$(LEN(I$)+1)="/"
630 IF PG THEN ? "SINGLE PAGES (1=YES,
  0=NO)":INPUT SPAGE
700 IF OP=3 THEN POKE 559,0:REM SHUT O
  FF DISPLAY TO SAVE TIME
705 IF OP=5 THEN GRAPHICS 0
710 LINE=0:POSITION 0,3:FL=0
724 IF B=4 THEN 2400
727 IF B=16 AND OP=3 THEN FOR I=1 TO P
  S-LINE:LPRINT " ":NEXT I:LINE=0:IF PG
  THEN GOSUB 880
728 IF B=16 AND OP=5 THEN ? :?" :?" :LIN
  E=0:IF PG THEN GOSUB 880
811 IF PEEK(764)<>255 THEN GOSUB 980
815 IF FF>0 OR PG THEN LINE=LINE+LINT:
  IF LINE>(PS-FF) THEN GOSUB 870
830 IF OP=3 THEN LPRINT SP$(1,SP):A$:I
  F LINT-1 THEN FOR I=1 TO LINT-1:LPRINT
  :NEXT I
840 IF OP=5 THEN ? SP$(1,SP):A$:IF LIN
  T-1 THEN FOR I=1 TO LINT-1:?"NEXT I
  850 IF FL AND OP=5 THEN GOSUB 980:GOTO
  500
855 IF FL THEN 500
```

```
870 FOR I=1 TO FF:IF OP=3 THEN LPRINT
875 ? :NEXT I:IF NOT (PG) THEN RETURN

880 PAGE=PAGE+1:SP=LL-(LEN(I$)+INT(LOG
  (PAGE))+6)+LM
882 IF OP=3 AND SPAGE THEN FOR I=1 TO
  20:SOUND 0,142,10,8:NEXT I:SOUND 0,0,0
  ,0:GOSUB 980
885 IF OP=3 THEN LPRINT SP$(1,SP):I$:"
  Page ";PAGE:LPRINT :LPRINT :GOTO 895
890 ? SP$(1,SP):I$:"Page ";PAGE:?" :?
895 LINE=3:RETURN
950 POKE 559,34:?"ERROR ";PEEK(195):"
  AT ";256*PEEK(187)+PEEK(186):?"PRESS
  RETURN TO CONTINUE":INPUT I$
960 CLOSE #3:GOTO 500
980 POKE 712,8:POKE 764,255
985 IF PEEK(764)<>255 THEN POKE 764,25
  5:POKE 712,0:RETURN
990 GOTO 985
2400 TP=TP+1:GOSUB 3000:LINT=TEMP:GOTO
  715
```

TinyText2 Change Listing 2 by Dale Lutz

```
515 UP=OP+1:IF OP=7 THEN OP=1
530 IF OP=6 THEN ? "TINYDOS"
560 POKE 764,255:POSITION 20,1:?"q":P
  OSITION 0,1:ON OP GOTO 2000,290,594,15
  00,594,4000
950 POKE 82,0:POKE 559,34:?"ERROR ";P
  EEK(195):" AT ";256*PEEK(187)+PEEK(186
  ):?"PRESS RETURN TO CONTINUE":INPUT I
  $
960 POKE 752,0:CLOSE #3:GOTO 500
4000 GRAPHICS 0:POKE ST-1,70:POKE ST+2
  ,7
4020 CLOSE #4:OPEN #4,6,0,"D:*.":COUN
  T=0:TRAP 4030:POKE 752,1:?"CHR$(125):P
  OSITION 20,0:?"TINYDOS"
4023 POSITION 0,2:POKE 82,1:?"FUNC=0
4025 INPUT #4,A$:COUNT=COUNT+1:GOTO 40
  25
4030 TRAP 950:CLOSE #4:OPEN #4,6,0,"D:
  *.*"
4032 FLAG=0:IF COUNT>26 THEN 4100
4035 FLAG=1:LIMIT=COUNT:BOTT=1:FOR I=1
  TO COUNT
4040 IF I-INT(I/26)*26=INT((LIMIT-BOTT
  )/2+1.5) THEN POKE 82,21:POSITION 0,2:
  ?
4045 GOSUB 4200:NEXT I
4047 POKE 82,1:IF COUNT>26 THEN POSITI
  ON 7,17:?"PUSH SPACE FOR NEXT SCREEN"
4048 POSITION 2,18:?"START, SELECT, S
  TART+SELECT, OPTION"
```

```
4050 POKE 764,255:FOR I=1 TO 30:NEXT I
  :IN=PEEK(53279):IF IN=3 THEN CLOSE #4:
  POKE 82,0:POKE 752,0:GOTO 500
4055 IF IN=6 THEN FUNC=1:POSITION 6,20
  :?" DELETE PUSH LETTER "
4060 IF IN=5 THEN FUNC=2:POSITION 6,20
  :?" LOCK/UNLOCK PUSH LETTER "
4065 IF IN=4 THEN FUNC=3:POSITION 6,20
  :?" RENAME FILE KEY TO START"
4070 I=PEEK(764):IF I=33 AND COUNT>26
  THEN ? CHR$(125):POSITION 20,0:?"TINY
  DOS":POSITION 0,2:?" :GOTO 4100
4075 IF I<255 AND FUNC THEN ON FUNC GO
  TO 4300,4340,4390
4080 GOTO 4050
4100 IF FLAG*26>COUNT THEN CLOSE #4:OP
  EN #4,6,0,"D:*.":FLAG=0
4110 FLAG=FLAG+1:IF FLAG*26>COUNT THEN
  4130
4120 LIMIT=FLAG*26:BOTT=LIMIT-25:FOR I
  =BOTT TO LIMIT:GOTO 4040
4130 LIMIT=COUNT:BOTT=FLAG*26-25:FOR I
  =BOTT TO COUNT:GOTO 4040
4200 INV=0:INPUT #4,A$:IF A$(1,1)="*"
  THEN INV=128
4205 IF I=COUNT THEN ? A$:RETURN
4210 ? CHR$(ASC("A"))-1+INV+(I-INT(I/26
  )*26)+26*(I=INT(I/26)*26)):A$(2)
4220 RETURN
4230 POKE 764,255
4235 IF PEEK(764)=255 THEN 4235
4240 GOTO 4020
4280 OPEN #5,4,0,"K":GET #5,IN:CLOSE #
  5:IF IN<65 OR IN>90 THEN POP :GOTO 404
  7
4290 ? "FILE ";CHR$(IN):" -- ";:RETURN

4300 GOSUB 4280:GOSUB 4500:IF LOCK THE
  N ? "FILE LOCKED...TYPE KEY TO CONTINU
  E":?" :GOTO 4230
4310 ? "Y TO CONFIRM, ANYTHING ELSE TO
  ABORT":?" :?" :POKE 764,255
4320 A=PEEK(764):IF A=255 THEN 4320
4325 IF A<>43 THEN 4020
4330 XIO 33,#4,0,0,A$:GOTO 4020
4339 REM LOCK/UNLOCK
4340 GOSUB 4280:GOSUB 4500:XIO 35+LOCK
  ,#4,0,0,A$:GOTO 4020
4389 REM RENAME FILE--NO FANCY STUFF,
  USER ENTERS IN WHOLE FILESPE
4390 POSITION 2,18:?"CHR$(156);CHR$(15
  6);CHR$(156);
4392 ? "PLEASE ENTER WHOLE FILESPE IN
  THE":?" FOLLOWING FORM--'D:orig.ext,n
  ew.ext'":POKE 764,255
4395 POKE 752,0:INPUT A$
4400 XIO 32,#4,0,0,A$:GOTO 4020
4499 REM ROUTINE TO EXTRACT FILESPE F
  ROM DIRECTORY (EXITS WITH A$ SETUP TO
  GO...)
```



```

4500 CLOSE #4:OPEN #4,6,0,"D:*. *"
4510 FOR I=1 TO FLAG*26-26+IN-64
4520 INPUT #4,A$:NEXT I
4530 LOCK=(A$(1,1)="*"):A=0
4540 A=A+1:IF A$(A+3,A+3)=" " THEN 456
0
4550 IF A<>8 THEN 4540
4555 I$=A$(3,10):GOTO 4565
4560 I$=A$(3,A+2)
4565 I$(LEN(I$)+1)="":I$(LEN(I$)+1)=A
$(11,13)
4570 A$="D:":A$(3)=I$:? A$:RETURN

```

TinyText2 Change Listing 3 by Dale Lutz

Change

Tiny Text change 3 listing
by Dale Lutz

```

515 REM DELETE THIS LINE
530 REM DELETE THIS LINE
560 REM DELETE THIS LINE
950 REM DELETE THIS LINE
960 REM DELETE THIS LINE
4000 GRAPHICS 0:ST=PEEK(560)+256*PEEK(
561)+4:POKE ST-1,70:POKE ST+2,7:DIM A$(
128),I$(128)
4020 CLOSE #4:OPEN #4,6,0,"D:*. *":COUN
T=0:TRAP 4030:POKE 752,1:? CHR$(125):P
OSITION 26,0:? "TINYDOS"
4030 TRAP 4020:CLOSE #4:OPEN #4,6,0,"D
:*. *"
4050 POKE 764,255:FOR I=1 TO 30:NEXT I
:IN=PEEK(53279):IF IN=3 THEN FUNC=4:PO
SITION 6,20
4052 IF IN=3 THEN ? "      RUN      PUS
H LETTER "
4070 I=PEEK(764):IF I=33 AND COUNT>26
THEN ? CHR$(125):POSITION 26,0:? "TINY
DOS":POSITION 0,2:? :GOTO 4100
4075 IF I<255 AND FUNC THEN ON FUNC GO
TO 4300,4340,4390,4600
4599 REM RUN FILE
4600 GOSUB 4280:GOSUB 4500:RUN A$

```

```

280 OPEN #1,4,0,FILENAME1$
290 GET #1,F
300 MODE=8:GOSUB 350:POKE SCRADDR,F:SC
RADDR=SCRADDR+1:SCRNUM=SCRNUM-1:GOTO 3
10
310 POKE 850,7:PICTURELOAD=USR(ADR(MAC
HINE$),16,SCRADDR,SCRNUM)
320 CLOSE #1
330 IF PEEK(53279)<>6 THEN 330
340 GOTO 470
350 GRAPHICS MODE+16:C=0:SCRADDR=PEEK(
88)+PEEK(89)*256
360 POKE 708,A:POKE 709,B:POKE 710,C:P
OKE 712,D:RETURN
370 FOR READR=1 TO 31
380 READ A
390 MACHINE$(READR,READR)=CHR$(A)
400 NEXT READR
410 RETURN
420 DATA 104,104,104,170,104,157
430 DATA 69,3,104,157,68,3,104
440 DATA 157,73,3,104,157,72,3
450 DATA 32,86,228,192,0,48,1,96
460 DATA 132,203,96
470 ? CHR$(125):SETCOLOR 2,0,0:POKE 75
2,1
480 POSITION 6,6:? "MICROPAINTER PICTU
RE LOADER"
490 POSITION 10,7:? "(C) 1983 CSOFTWAR
E"
500 POSITION 12,8:? "BY SHANE ROLIN"
510 POSITION 4,10:? "SELECT FUNCTION A
ND PRESS START"
520 POSITION 11,20:? "FILENAME=";FILEN
AME2$
530 POSITION 5,13:? "LOAD A PICTURE(EN
TER NEW FILENAME)"
540 POSITION 5,14:? "DISK DIRECTORY"
550 POSITION 5,15:? "DELETE A FILE(OR
PICTURE)"
560 POSITION 5,16:? "QUIT"
570 POKE 53279,0
580 POSITION 2,XY-1:? "  "
590 POSITION 2,XY+1:? "  "
600 IF XY=17 THEN XY=13
610 IF XY=12 THEN XY=16
620 POSITION 2,XY:? "--";CHR$(27);CHR$
(159)
630 IF PEEK(53279)=7 THEN 630
640 FOR T=1 TO 50:NEXT T
650 IF PEEK(53279)=5 THEN XY=XY+1
660 IF PEEK(53279)=6 THEN 690
670 IF PEEK(53279)=3 THEN XY=XY-1
680 GOTO 570
690 IF XY=13 THEN 740
700 IF XY=14 THEN 850

```

```

710 IF XY=15 THEN 950
720 IF XY=16 THEN ? CHR$(125):POKE 710
,149:POKE 752,0:END
730 GOTO 470
740 ? CHR$(125):FOR T=1 TO 20:FILENAME
2$(T,T)=" ":FILENAME1$(T,T)=" ":NEXT T
750 FILENAME1$="D:"
760 POSITION 7,7:? "LOAD A PICTURE"
770 POSITION 0,10:? "INPUT FILENAME ON
LY(NO 'D:' IS REQUIRED)"
780 POSITION 7,12:INPUT FILENAME2$
790 ? :? "NEW FILENAME IS D:";FILENAME
2$
800 ? "IS THIS CORRECT(Y/N)"
810 GET #2,A
820 IF CHR$(A)="Y" THEN 260
830 IF CHR$(A)="N" THEN 470
840 GOTO 810
850 ? CHR$(125)
860 CLOSE #3
870 OPEN #3,6,0,"D:*. *"
880 TRAP 920
890 INPUT #3,DIRECT$
900 ? DIRECT$
910 GOTO 880
920 POSITION 25,10:? "PRESS START":TRA
P 470
930 IF PEEK(53279)<>6 THEN 930
940 CLOSE #3:GOTO 470
950 ? CHR$(125)
960 POSITION 14,7:? "DELETE A FILE"
970 POSITION 0,10:? "INPUT FILENAME ON
LY(NO 'D:' IS REQUIRED)"
980 POSITION 7,12:INPUT FILENAME2$
990 FILENAME1$(3,19)=FILENAME2$
1000 ? :? "FILE TO ERASE IS ";FILENAME
1$
1010 ? "IS THIS CORRECT(Y/N)"
1020 GET #2,A
1030 IF CHR$(A)="Y" THEN XIO 33,#1,0,0
,FILENAME1$:GOTO 470
1040 IF CHR$(A)="N" THEN 470
1050 GOTO 810

```

Listing 2 next month

NOTE: The listings of this issue were printed by the new Brother Mode 50 daisywheel typewriter with a Brother IF-50 interface. This is a very nice inexpensive (about \$400) typewriter that is slow (12cps) but works well. The ¶ means ESC-CLEAR. The typewriter used is an English Quadro 15, reduce about 25% in size.

-M.D

PMHELPER 2.0

INTRODUCTION

By Stan Ockers

Player Missile Graphics is a very useful technique when a game requires motion of images or animation. Unfortunately, quite a bit of setup is necessary and vertical movement is complex enough to demand assembly language programming. What if there was a program a person could use to help design PM images with a joystick, assign them to certain players, choose the colors, sizes and initial positions of players and save the result to disk or tape? PMHELPER 2.0 is such a program. The finished product can be saved in LIST format so it can be merged with a new program. Also included in the saved version are lines necessary to implement PM graphics and a Vertical Blank Interrupt routine which reduces motion of players (either horizontal or vertical) to POKEs in specified locations. Images can also be changed just as easily using POKEs.

TYPING IT IN

Listing 1 should be typed in noting the inverse character remarks in a couple of places. The program will work as listed but the initialization takes quite some time because of the string initialization. This slowness will also follow through to the listed module. The process can be speeded up considerably by defining strings directly in the program. It wasn't done in the listing because it makes it very difficult to print the listing or ENTER the result. You can however change the program once it is ENTERed from listing 1.

First ENTER the program as listed, make a copy and then RUN the program. Hit RESET to interrupt the program. Now type "? INVBIS" (omit the quotes). A number of characters will be printed. Enclose them in quotes, then use the CTRL-INSERT keys to move the characters to the right. Add "10300 DIM INVBIS(10):INVBIS=" at the beginning and hit RETURN. You have now replaced line 10300 and line 10330 is now unnecessary so you may delete it. In a similar manner you can replace lines 10310 and 10340 by defining OUTVBIS directly. The same can be done with MVPMS, deleting lines 10050-10070 and replacing 10000 with "DIM MVPMS(48):MVPMS=" appropriate characters.

INVBIS is more complicated. Not only is it too long, it also contains characters which cause odd cursor movements when printed. To transform it first type in: "?VBIS(1,4);" "?VBIS(6,46);" "?VBIS(48,53);" "?VBIS(55,64);" "?VBIS(66,72);". This puts blank spaces where the troublesome characters occur. Insert "10100 DIM VBIS(124):VBIS=" at the beginning and leave an extra blank spot at the end when enclosing the string in quotes. Now you must fill the blank spots with the appropriate characters, using the CTRL-Arrow keys to position the cursor over them. The first filling at string position 4 uses the sequence ESC-CTRL-UP ARROW and produces an inverse upward pointing arrow. All other spaces are to be filled with the ESC-CTRL-DOWN ARROW sequence which produces an inverse downward pointing arrow. These occur at string positions 47,54,65 and 73 which is the last blank you inserted. Press RETURN to enter this first portion of the string as line 10100. The second part of the string can be produced with: "?VBIS(74,92);" "?VBIS(94,99);" "?VBIS(101,115);" "?VBIS(117,118);" "?VBIS(120,124)". Blanks at string positions 93,100,116 and 119 should be filled using the ESC-CTRL-DOWN ARROW sequence as before. Use it in line 10110 with "10110 VBIS(74)=", followed by the string in quotes. You may get rid of lines 10150-10200. All remarks can be deleted in this "running" version. These changes should improve the initialization time both for PMHELPER 2.0 and for programs using the PM module produced by the program.

USING THE PROGRAM:

PMHELPER 2.0 is set up for 12 different images (0-11), any one of which can be assigned to four different players (0-3). A sketch area of 8X20 pixels allows designing images using a joystick in PLYR 1 port. Pressing the trigger will toggle a pixel between "on" and "off". The numbers 0 and 1-3 chooses the player, and the player number is indicated directly below the sketch area. A number of values corresponding to attributes of the chosen player appear in the bottom of the text area.

You can use the "+" and "*" keys to select values you want to modify and the "-" and "=" keys to change the values. You will see the image of the chosen player change when changing values. X and Y positioning allows superposition for multicolored images. The width of players can also be chosen and will be "remembered" along with color and luminance values. Any players you are not interested in can be eliminated by setting their X values to position them offscreen. The image in the sketch area is not necessarily associated with any player until the spacebar is pressed. When space is pressed, the sketch image will replace an image in memory, the one associated with IM-

AGE#. Since this image is associate with the present player, it will change also. CLEAR can be used to blank the sketch area and INSERT can be used to transfer the image given by IMAGE# to the sketch area.

When you are satisfied with the images you have created, press "S" for SAVE. You will be asked for a filename for saving the work. I use names like "D:NAME.LST" because files are SAVED in LIST format for addition to another program using ENTER. If you are using cassette, use "C1". When you press RETURN you will see some lines listed on the screen and some cursor movement. The program is using the screen to ENTER new lines before the SAVE. It will return to the program when finished. You will also see some garbage from the PM images. Ignore it, your initial PM images will be restored when the program re-initializes. If the program stops with an error message, simply re-run it. Your images will return. If you want to modify a previously listed image set, stop the program with RESET, ENTER the desired image set and RUN the program.

USING THE PM MODULE

The modules created with PMHELPER 2.0 can be used with very little extra code. The minimum amount appears as follows:

```
100 GOSUB 10000
110 GRAPHICS 0
120 POKE 53277,3:POKE 559,62
130 STOP
```

Line 100 goes to the module to set up PM graphics and insert your images. Of course you must have ENTERed the module in the program. Line 110 represents putting a desired background on the screen; any graphics mode changes have to be done before PM graphics is hooked in by line 120. If the graphics mode is reset after line 120 the PM images smear out over the whole vertical length of the screen.

X and Y positions of the players can be controlled with POKEs. To assist you, the variables XPO0 through XPO3 and YPO0 through YPO3 have been set up. These variables DO NOT HOLD THE VALUES OF X AND Y POSITIONS, they hold the LOCATIONS where these values are stored. To find the Y position of player two you PEEK(YPO2). To change the Y position of player two, POKE YPO2,# where # is a number between 0 and 255. The code to move player 2 back and forth across the screen might look as follows:

```
130 DELTA = 1
140 P = PEEK(XPO2):IF P>200 THEN DELTA = -1
150 IF P< 50 THEN DELTA = 1
160 POKE XPO2,P + DELTA:GOTO 140
```

Movement doesn't have to be to adjacent positions, but can be anywhere on the screen. Try changing the DELTA's above to 2, 3 or more.

Image changes are also handled by POKEs and the variables IMG0-IMG3 are set up to help. again, these variables refer locations where image numbers are stored, not to the image numbers themselves. They represent the storage locations for the image numbers assigned to each of the four players. The numbers are not simply 0-11 representing the twelve images but must be multiplied by 20. For example, to make player one take on image six, POKE IMG1,120. Animation can be handled by rotating image numbers. If images 7 through 11 hold various figures of a man running, you could make him run in place with:

```
130 I = 120:SPEED = 30
140 I = I + 20:IF I>220 THEN I = 140
150 POKE IMG2,I
160 FOR J = 1 TO SPEED:NEXT J:GOTO 140
```

Line 160 is just to adjust the speed and can probably be replaced by other code which gives horizontal movement like the previous example.

LIMITATIONS

The program does not include missiles and is limited to single line resolution. The variable PMPG holds the page where the PM area begins so you can write your own routines to handle missiles if you wish. The missile area starts at PMPG*256 + 768, (PMPG + 3 pages). The routine reserves 8 pages at the top of memory for the PM area. The lowest page of this area is not used for anything so it make a good buffer to allow for those situations where the program overruns the top of memory. For this reason, PM modules should be the last called of those which reserve memory. The one page buffer will allow for most situations, but not one. If the text area is allowed to scroll, it can move things around for more than two pages above RAMTOP. Since the images are located at the second page, they will be messed up.

The program is set up for 12 images, each 20 bytes long. You could double this by drawing two images in one sketch area. The numbers POKEd into IMG0-IMG3 will be IMAGE# * 10 and the image length values in page 6 (bytes 1688-1691) will have to be set to 10. Other combinations are possible because the number POKEd is just a offset from the beginning of the image storage area.

—Stan Ockers

VARIABLES IN PMHELPER 2.0

MVPMS: Holds routine making room for PM graphics at the top of memory
 VBIS: Holds Vertical Blank Interrupt routine permitting movement and image changes with POKEs
 SS: Used to hold string of inverses characters indicating selected variable
 INVBIS: Holds routine to insert VBI routine
 OUTVBIS: Holds routine to disengage VBI routine
 F\$: Filename of module
 COLR(0)-COLR(3),LUM(0)-LUM(3),XPO(0)-XPO(3),YPO(0)-YPO(3),IMG(0)-IMG(3): Values of color, luminance, X position, Y position, width and image numbers of four players for sketching part of program.
 XPO1-XPO3,YPO1-YPO3,IMG0-IMG3: Locations to be POKEd with corresponding values of player X position, Y position and image numbers for players
 IM(0)-IM(19): Numbers representing image in sketch area
 C9: Color-luminance combination
 SEL: Player variable selected for changing
 PLYR: Player number
 PMPG: Page where player missile area starts
 X,Y,Z: Horizontal position, vertical position and pixel color under cursor
 S: Joystick value
 KEY: Value of keyboard key pressed
 VBI, HI, LO: Used in determining address of VBI routine

PAGE 6 LOCATIONS USED IN PMHELPER 2.0

1664-1667: OLD X POSITIONS PLAYERS 0-3
 1668-1671: NEW X POSITIONS PLAYERS 0-3
 1672-1675: OLD IMAGE ASSOCIATED WITH PLAYER
 1676-1679 NEW IMAGE ASSOCIATED WITH PLAYER
 1680-1683: OLD Y POSITION PLAYERS 0-3
 1684-1687: NEW Y POSITION PLAYERS 0-3
 1688-1691: SIZE OF PLAYERS 0-3
 1692: PAGE WHERE PM AREA STARTS
 1693: TEMPORARY COUNTER LOCATION

COMPUTER CUBE

Program Description

LINES:

10 OPENS the keyboard for input.
 100-105 Redefines the required characters and duplicates the rest of the character set.
 120 Finds the memory position of the first character on the first line of the screen. I use this as a point of reference as I POKE the characters on the screen. This is much faster, and I only need one variable to store the coordinates of a block in the cube.
 200-212 Set up the initial screen display and put up the title.
 215-216 Determine if a key or
 is pushed and loop if not.
 220-224 Check for valid key input and loop if not.
 226-231 Print appropriate face name for key pushed.
 240-241 Print the chosen direction CL or AC.
 245 Goes to subroutine to rotate the chosen face.
 300 Randomly chooses a subroutine to mix the cube. This is repeated as many times as you select (3-20).
 310-469 Contain the above mentioned subroutines. Their operation should be reasonably clear. They just shift the values of blocks which will be moved when that face is rotated.
 900-910 Set the correct values to reset the cube if you give up.
 1000 Data for the new characters.

INSTRUCTIONS:

First, select the number of random moves you wish the computer to make. Push

SELECT

and the number appears at the bottom left corner of the screen.
 Press

START

when ready to play and the computer will mix up the cube. Now press the key to select the face: B = back; T = top; D = down; F = front; L = left; R = right. The diagram at the top shows this and also which face is which.

After the name of the face appears at the bottom of the screen, press the key for the direction: A = anti-clockwise; C = clockwise. Now the computer will move the face for you and wait for new instructions. Press

START

at any time to give up.

Remember: type the program exactly as listed, except for REMS.
 — Sydney Brown

ASTROLOGY

ASTROLOGY: A LOOK INTO THE FUTURE

by SAM ROBERTS; ELCOMP Publishing Co. Book: #171, \$9.95; Book & Disk: #7223, \$29.95

For those of you interested in exploring the world of astrology, without a large investment in software, this book may be just what you need.

The book is intended to be complete in itself without buying the optional program disk. But it is not really so simple, because the disk contains utilities not in the book.

Part one of the book is a brief view of astrology in general. There are many excellent books for the beginner in your local bookstore or library. I suggest you skip part one and get one of these. Part two contains a program listing which you can type in and use to calculate birth charts for your families and friends.

The program is the interesting part of the book. It takes a bit to type it in (6 half-pages of code), but with patience and/or typing skill it shouldn't prove too difficult. Debugging might be difficult for someone without experience in both computing and astrology. But if you type it in exactly as in the book it should run fine, right? Well, yes and no.

The graphics part of the program, which will print the wheel to the screen, uses a custom character set for the signs of the Zodiac and the symbols for the planets. They provide the data for the characters in the back of the book. To load the character set into memory, use the L option in DOS to load ASTRO.BIN. Without the character set in memory, you can still calculate a chart, but when you try and draw the wheel on the screen it will be empty.

Now to cast some charts! To calculate a birthchart the following information is needed: 1) the date and year of the person's birth; 2) a verifiable, accurate time of birth, and; 3) the location of birth (latitude, longitude, and time Zone). A birth certificate, family or hospital records will usually supply the date and time. An atlas will supply the coordinates. You have to be very careful about daylight savings time and time zone changes — the only way to assure accuracy is by referring to a book titled "time changes in the U.S.", by D.C. Doane. Other than this the data is fairly easy to aquire.

Once you have the data gathered and ready, load the character set and RUN the main program. The program will prompt you to type in the birth data: you have to be very careful of the format of entry — for example, a latitude of 123 degrees, 4 minutes, and 18 seconds (Eugene, OR) has to be entered 123.0418. The input routines are primitive and some of the prompting is ambiguous. There is no real error checking and the program is easily crashed. Worse yet, you could unknowingly input the wrong data and get an inaccurate chart. It's a good idea to test the program with a chart from a book to have something to compare against. If the results are within a degree or so all around then you're ok.

Assuming you've got the data in correctly, you will be rewarded with an accurate listing of the signs and degrees of the 12 house cusps (see your astrology book for definitions). The listings are in 360 degree notation, which means you have to transpose them to Zodiac note to get the sign involved. Each sign is 30 degrees. So Aries is from 0 to 29 degrees, Taurus is from 30 to 59, and so on to Pisces, 330 to 359. So 347 degrees is 17 degrees of Pisces (347 — the sign). After a while this will come naturally.

When ready to continue, hit return and you will have a listing of the position of each planet (Sun and Moon included) at the time of the person's birth. Hit return again for a tabulation of aspects, or angular relationships between the planets. Then hit return a final time, and the program will begin to plot all this data onto a chart on the screen. It draws very slowly (GR.8), but it is one of the nicest computer charts I've seen. There is a program to dump to an Epson printer on the disk, but not in the book. I imagine any screen dump utility will do as well.

What we have here is a low priced program to allow you to begin to discover astrology for yourself — what it can and cannot do. If you want to "tell the future", as the title implies, save your money. The accuracy of the calculations is adequate for the average user. It is not an easy program to use, but once you've figured it out you will be able to generate all the charts you like. If you plan to buy the disk, there are other more versatile and powerful programs now available for the Atari from Matrix Software, 315 Marion Ave., Big Rapids, MI 49307. I strongly endorse these for the serious astrologer.

As both an astrologer and computer enthusiast, I can appreciate the effort involved in writing good astrological software. I can't escape the feeling, however, that this package needed a bit more polishing before being released on the general market.

—Jiva

I invite anyone with an interest in astrological software, research to share, or questions, to write me at: 1476 S. Brooklyn, Eugene, OR 97403

BENIOFF AT LARGE

LOOKS AT: ATARI 1200XL

A few months ago, when ATARI released the 1200XL, I ran out and bought one. The cost of the computer was \$899.00. When I made the purchase, I knew I was going to have some software problems. I knew it might not be the computer of the future. But I was looking for a change.

Whenever I mention I own a 1200, I always hear snickers and smirks. People say things like, "Why did you buy it?" This article is intended to clear up a lot of the confusion and myths of the 1200. And, it is well warranted — because the new machines (600XL, 800XL, 1400XL, 1450XL) will have the same operating system. Yes, you heard me right. Benioff goes in search of: ATARI 1200XL.

What about software? This question is an important one. Yes, there are some things which do not work on the 1200. For example, most of Synapse's older versions do not work. Most word processors do not work, including the ATARI word processor! Most third party carts do not work (excluding ROMOX). So, what is a poor 1200 owner to do? Well, it is not as bad as it seems. I use the ATARIWRITER. It is the new ATARI word processing program. ATARI exchanged it with me for my old ATARI word processor. It is better than any word processor written so far. Of course, I still can't use a spelling checker. Spell Wizard doesn't work, and Datasoft keeps trying to correct the problem, and can't. I think about 10% of the current software doesn't work. The key here is: companies are currently changing their software to work on the 1200. So, software is a problem, but it is currently being corrected.

Let's take into consideration the good parts of the 1200. These include the excellent keyboard. I think it is like typing on feathers. It is leaps and bounds above the 800s. The terrific changes in the operating system. The addition of a "help" key. The recessed function strip at the top of the keyboard. The elongated shift and delete keys. All of the wires hanging directly out of the back. This makes the 1200 a superior machine to the 800.

Would I buy it again? Yes. Of course, I would only have to pay \$299.00. This price is terrific. I want to thank Dennis Wallin at Datasoft for trying to get me a Spell Wizard to work on the 1200. If he does, I will let you know in the next issue.

I am going back to school tomorrow, so I may have an erratic schedule. If not, an erotic schedule.

— Marc Benioff
31 West 28th Street
Los Angeles, CA 90007

BRYAN'S ARCADE

This month I received two new games, both on ROM cartridges. The games are Mr. Cool from Sierra On-Line and the second is Flapper, a new game from ROMOX written by the great reviewer Marc Benioff.

Mr.Cool is a Q*Bert type game. The object of the game is as follows. You are an ice cube and your mission is to make the "Hot Plates" cold. You make them cold by jumping on them and changing their color. You have to keep jumping on them until they are the same color as the color at the top of the screen. On certain screens you have to jump on them three or four times until they get to be the color you want them to be.

Obstacles include hot springs which fly across the screen and if they barely touch our hero Mr.Cool, he'll instantly melt. There are two other obstacles. One is another type of hot spring which jumps down the pyramid of ice cubes and tries to make Mr.Cool melt. Your only other threat is the bonus which runs down at an amazingly fast rate and when it gets to zero you will instantly melt.

Overall Mr.Cool is a great game and a pretty good Q*Bert type game. The sound effects aren't too good, but the graphics are pretty good. There is room for improvement. I recommend Mr.Cool to anybody who likes Q*Bert in the arcades it's definitely worth the money.

Flapper is an interesting and totally original game designed by Marc Benioff for ROMOX, Inc. You are Flapper. Flapper is a little blue and pink bird whose goal is to "pop" the fruit with a jack straw. There are four rows of fruit with five in each row. Flapper has to pop all of the fruit in each row while avoiding the Sprinklers and Aggrobots (I didn't think up the names). You have to pop all of the fruit before the sun goes down.

Flapper has excellent music which was written by Marc's grandmother. Russ Wetmore programmed it (the same guy who wrote and did the music for Preppie and Preppie II). That's why the music sounds so good. The graphics are pretty good, you can even see Flapper's wings move when he flies.

Well, that's all for this month! Gotta go,

— Bryan Dunn
Games Editor

COMMUNICATIONS

Information recovery is an important part of my business as a private investigator. When people inquire how I recover so much information for my clients, I respond "through my computer." Where else? Sure, there is a lot I need to know in fact finding. But through computer research I can cut the time of search, cutting the cost of my services.

I have over 366 databases available to my office through my modem. One of the more recent discoveries is the well-known Telex and TWX services.

Western Union has offered this service since the days of early teletype. Now I can get high-speed, low-cost link to over 1.5 million concerns world-wide through a new Western Union service called "Easylink Service".

Clients may inquire what type of computer I use in my offices. I hesitate. Atari, of course! I have two Atari's in full function. By utilizing my Hayes Smartmodem and Datasoft's "Tele-talk" modem program, I can access any database out there. Western Union has an extensive Telex directory and Buyers' Guide listing other subscribers. Off peak transmission is available at significantly reduced rates. There is no minimum usage and no long-term commitments with Western Union. Now I can send Mailgrams, Telegrams and Cablegrams anywhere.

The service also has a high-speed store and forward message service. Through their "Electronic Mailbox" they will hold messages until you call and retrieve them. The best part, of course, is the toll free number you can use to access their system. If you desire you can get on the auto-receive mode to receive responses or messages immediately. In order to use the system, all you need is modem, software, computer, printer (optional). The software must be able to accept even parity, seven data bits and one stop bit. Basically what all this means is a good program and modem.

Most of the programs out there will work. I have experienced problems with Telelink I, however. The initial cost to sign up for this service is \$35. The monthly charge is also \$35. There is also a cost for the actual calls. I recommend this type of service is you use or are interested in expanding your communication potential. If you want more information, call your local Western Union office.

— Dennis Hettman

CLOAD ERRORS

I am another new Atari owner who was plagued with excessive CLOAD errors. Armed with a copy of the April, 1983 issue of Antic, I added several test points in an effort to determine where the actual failure occurred. The data out-line appeared to have an excessive amount of jitter when viewed on a dual trace oscilloscope along with the output of the active filters.

I modified the diode/RC networks between the active filters and the comparator as a possible cure, after checking the DC regulator to the motor drive for ripple.

The attached diagrams showing the 410 circuitry show my modifications. I have had NO CLOAD errors since.

To insure it was not a cold solder joint or other problem I may have corrected by the addition of test points, I obtained another 410 with similar problems. After doing the simple modification and alignment, its operation also significantly improved. It was an older model 410 which had a different parts layout, but the circuitry was the same. The owner of this machine has not reported a CLOAD error since.

While this modification is not much more difficult than the resistor replacements, the alignment requires test equipment not normally found in the home. Any competent stereo service shop can do the alignment in a few minutes time. Certainly no more than an hour.

Atari 410 Modification & Alignment Procedure:

This adjustment dynamically sets the signal circuit to the mid point between the mark and space frequencies used.

1. Remove R-19 (330 K ohm);
2. Choose a suitable location on the PC board and install either a 500 K ohm mult. turn pot., or any series combination of fixed resistor and pot. which will give a range from approximately 250 K ohm to 350 K ohm;
3. Connect this pot. or combination directly in place of the removed R-19;
4. Set the pot. to maximum resistance
5. Connect a DC coupled oscilloscope or high Z voltmeter to the data out line (pin 3) and ground;
6. Connect an audio oscillator set at 4660 Hz to the limiter amp input and ground;
7. Condition the 410 for play mode — this can be done by pressing the PLAY button and POKEing 54018,52 or by applying a +5V to the recorder motor on line (pin 4);
8. Adjust the potentiometer until a DC transition, or instability, occurs on the data out line.

— Stephen E. Matern
Vice-President, Bonneville Atari Group

TYPESETTING FROM YOUR COMPUTER

ATARI OWNERS: If you have a modem, text editor, and communications program to send ASCII files, you should consider the improved readability and cost savings provided by **TYPESETTING** your program documentation, manuscript, newsletter, or other lengthy text instead of just reproducing it from line printer or daisy-wheel output. Computer typesetting by telephone offers you high quality, space-saving copy that creates the professional image you want! Hundreds of type styles to choose from with 8 styles and 12 sizes "on line." And it's easy to encode your copy with the few typesetting commands you need.

COMPLETE CONFIDENTIALITY GUARANTEED

— Bonded for your protection —

PUBLICATION DESIGN, EDITING, & PRODUCTION

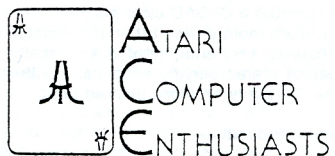
Editing & Design Services
Inc.

30 East 13th Avenue Eugene, Oregon 97401
Phone 503/683-2657

MEETING!
Weds Sept 14, 7:30

Forum #308 LCC

Door Prizes donated by
Computer Tutor



3662 Vine Maple Dr. Eugene OR 97405

FIRST CLASS MAIL

Atari Computer Enthusiasts

A.C.E. is an independent, non-profit and tax exempt computer club and user's group with no connection to the Atari Company, a division of Warner Communication Company. We are a group interested in educating our members in the use of the Atari Computer and in giving the latest News, Reviews and Rumors.

All our articles, reviews and programs come from you, our members.

Our membership is world-wide; membership fees include the A.C.E. Newsletter. Dues are \$10 a year for U.S., and \$20 a year Overseas Airmail and include about 10 issues a year of the ACE Newsletter.

Subscription Dep't: 3662 Vine Maple Dr., Eugene, OR 97405.

President—Kirt Stockwell, 4325 Sean , Eugene, OR 97402 / 503-689-5355

Vice Pres—Larry Gold, 1927 McLean Blvd., Eugene, Or 97405 / 503-686-1490

Secretary—Charles Andrews, POB 1613, Eugene, Or 974401613 / 503-747-9892

Librarian—Ron and Aaron Ness, 374 Blackfoot, Eugene, Or 97404 (503)689-7106

Editors—Mike Dunn, 3662 Vine Maple Dr., Eugene, Or 97405 / 503-344-6193

Jim Bumpas, 4405 Dillard Rd., Eugene, Or 97405 / 503-484-9925

E.R.A.C.E. (Education SIG Editor)—Ali Erickson, 295 Foxtail Dr., Eugene, Or 97405 / 503-687-1133

E.R.A.C.E. Corresponding Secretary—Robert Browning, 90 W. Myoak, Eugene, OR 97404 (503) 689-1513

Send 27¢ in stamps to the Ness' for the new, updated ACE Library List!! (or 50¢ Overseas)

Bulletin Board

(503) 343-4352

On line 24 hours a day, except for servicing and updating. Consists of a Tara equipped 48K Atari 400 with a TARA keyboard, 4 double-density disk drives with the ATR 8000, an Atari 825 printer, a Hayes SmartModem; running the ARMUDIC Bulletin Board software written by Frank L. Huband, 1206 N. Stafford St., Arlington, VA 22201. See the Nov '82 issue for complete details.